# E-LEARNING SYSTEM BASED ON CLOUD COMPUTING

By
KAPIL GARG - 101279

Under the Supervision of
DR. DEEPAK DAHIYA

MAY 2014

*Project Report Submitted in partial fulfillment of the requirement for the Degree of*
BACHELOR OF TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
JAYPEE UNIVERSITY OF INFORMATION TECNOLOGY,
WAKNAGHAT

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY WAKNAGHAT**
**SOLAN , HIMACHAL PRADESH**

**Date:** ____________________

# CERTIFICATE

This is to certify that the work titled **E-LEARNING SYSTEM BASED ON CLOUD COMPUTING** submitted by **KAPIL GARG** in partial fulfillment for the award of degree of Bachelor of Technology (C.S.E.) of **JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,WAKNAGHAT** has been carried out under my supervision . This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature in full of Supervisor: ________________________________________

Name in Capital Block Letters: **Dr. DEEPAK DAHYIA**

Designation: **Professor , Department of Computer Science and ICT ,JUIT**

# ACKNOWLEDGEMENT

No venture can be completed without the blessing of Almighty. I consider it my bounded duty to bow to Almighty whose king blessing always inspire us on the right path of the life.

Science has caused many frontiers so has human efforts towards human research. My guide **Dr. Deepak Dahyia** , Professor , Department of Computer Science and ICT, JUIT, has indeed acted as a light house showing me the need of sustained effort in the field of Cloud Computing to learn more and more. So I take this opportunity to thank him, for lending me stimulating suggestions, innovative quality guidance and creative thinking. He provides me the kind of strategies required for the completion of the task. I am grateful to him for the support, he provides me in doing thinks at my pace and for being patient for my mistakes.

Signature of the Student: ____________________

Name of Student: KAPIL GARG

Date: ____________________

# ABSTRACT

E-learning system need to improve its infrastructure, which can devote the required computation and storage resources for e-learning systems. The objective of the project is to combine various technologies to design architecture which describe E-learning systems. E-learning is the computer and network-enabled transfer of skills and knowledge. E-learning applications and processes include Web-based learning, computer-based learning, virtual education opportunities and digital collaboration.

# Table of Contents

# CHAPTER 1: INTRODUCTION

E-LEARNING SYSTEM need to improve its infrastructure, which can devote the required computation and storage resources for e-learning systems. Microsoft cloud computing technologiesal though in their early stages, have managed to change the way applications are going to be developed and accessed. The objective of the paper is to combine various technologies to design architecture which describe E-learning systems. Web 3.0 uses widge, intelligent retrieval, user interest modeling and semantic annotation. These technologies are aimed at running applications as services over the internet on a flexible infrastructure. Cloud computing provides a low cost solution to academic institutions for their researchers, faculty and learners. In this paper we integrate cloud computing as a platform with web 3.0 forbuilding intelligent e-learning systems.

## 1.1 OVERVIEW

E-learning is the computer and network-enabled transfer of skills and knowledge. E-learning applications and processes include Web-based learning, computer-based learning, virtual education opportunities and digital collaboration. Content is delivered via the Internet, audio or video tape, satellite TV, and CD-ROM. It can be self-paced or instructor-led and includes media in the form of text, image, animation, streaming video and audio. It is commonly thought that new technologies can make a big difference in education. Especially, children can use the huge interactivity of new media, and develop their skills, knowledge, and perception of the world, under their parents' monitoring, of course. Many proponents of e-learning believe that everyone must be equipped with basic knowledge in technology, as well as use it as a medium to reach a particular goal. E-learning is widely used today on different educational levels: continuous education, company trainings, academic courses, etc. There are various e-learning solutions from open source to commercial. There are at least two entities involved in an e-learning system: the students and the trainers. Some benefits of e-learning are discussed below: Time: One of the key benefits of online study is that one can learn or take a course through e-learning at any time as it is convenient for them. Podcasts and downloadable lectures mean that students are no longer constricted by a conventional timetable of lectures. Location: Neither are students restricted by their physical location. With an Internet connection, they can attend live online tutorials, participate in dedicated discussion forums or download course material and notes regardless of where they are. Communication: Another key advantage of online study is that it encourages and enables students to collaborate and communicate with their fellow students as well as their tutors.

## 1.2 Objective And Scope Of Project

This project is based on development of a system where users registered as students are able to use the system to access their study material and give tests and users registered as teachers/faculty are able to upload their material. It can be extended for uploading assignments, marking grades and evaluating students (for faculty).

In order for a needs assessment to have a successful outcome, it must accomplish many things--improving performance being just one. It must also

- determine how to achieve the high-level goals of the organization (such as increasing sales and fostering innovation)
- determine what system obstacles (other than training) need to be removed
- point to an intervention that will balance the conflicting needs of different stakeholders (IT versus HR, participants versus managers, budget versus vendor costs, and so on)
- pave the way for a new program.

Organizations implementing e-learning programs need to expand the usual needs assessment process by creating a high-level requirements document that includes

- objectives (macro organizational objectives and micro target learner population objectives)
- an e-learning readiness score
- a list of advantages and potential obstacles to e-learning adoption
- a list of possible e-learning configurations.

## 1.3 Benefits of Project

A flexible and convenient tool used to engage teachers and learners.

• It is ubiquitous in environment.

• No need to store and backing up it from one device to the other.

• Allows students to avail the E-learning resources from different places and can also access through different resources such as mobile, laptop and desktop computers provided that internet access is available.

• Students can experience a richer and diverse learning environment.

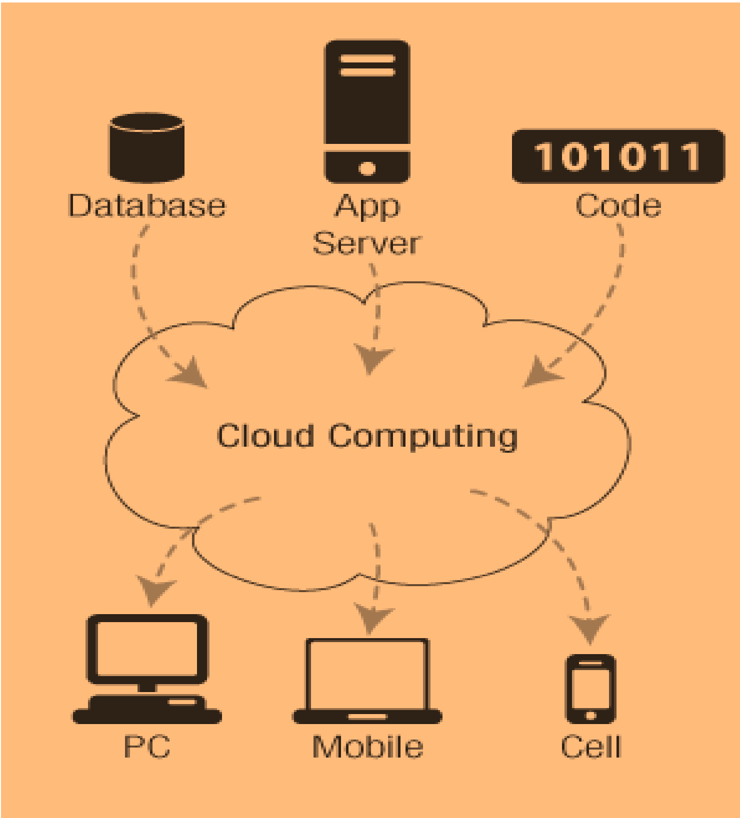
• It provides affordable solutions to the academic institutions.

• Flexible and dynamically scalable infrastructure, which reduces the overall maintenance cost.

• A virtualized e-learning solution which increases the performance and reduces overall cost and burden for the academic institution

# CHAPTER 2 : Project Review

## 2.1 Cloud Computing Background Study




Cloud Computing is a new paradigm that provides an appropriate pool of computing resources with its dynamic scalability and usage of virtualized resources as a service through the Internet. The resources can be network servers, applications, platforms, infrastructure segments and services. Cloud computing deliver services autonomously based on demand and provides sufficient network access, data resource environment and effectual flexibility. This technology is used for more efficient and cost effective computing by centralizing storage, memory, computing capacity of PC's and servers. With the tremendous advantages of cloud computing, we expect this technology to revolutionize the field of e-learning education. Cloud computing applications provide flexibility for all

Cloud is about a new business model for providing and obtaining IT services. Cloud computing promises to cut operational and capital cost. It lets IT departments focus on strategic projects rather than on managing their own datacenter. Cloud computing is building upon a number of ideas and principles, that have been established in the context of utility computing, grid computing, and autonomic computing a couple of years ago. However, in contrast to previous approaches, cloud computing no longer assumes that developers and users are aware of the provisioning and management infrastructure for cloud services.

Cloud computing has established the notion of a service as basic unit of abstraction. Services are "running in the cloud" and can be consumed via standard WebService interfaces. Services can be on different levels of abstraction - comparable to standard applications (Software-as-a-Service - SaaS), comparable to frameworks and programming platforms (Platform-as-a-Service - PaaS), or comparable to an (virtualized) IT infrastructure (Infrastructure-as-a-Service - IaaS). From the standpoint of a programmer, cloud computing on each of these levels of abstraction has a number of technical design choices, which impose implications on developers.

Cloud computing environments can be offering their services via the Internet - thus forming the public cloud - or can be offering services only on a company's intranet - effectively implementing a private cloud. The usage of cloud technology with services both on the intranet and the Internet simultaneously is sometimes called "hybrid cloud".

Cloud computing is appealing for certain kind of applications, namely those imposing a variable load and needing massive scaling (such as Web 2.0 apps during peak hours), those with a short or unpredictable lifetime, and those doing parallel computing using huge amounts of resources. Cloud computing also can be seen as a system consolidation approach to be used to handle mergers and acquisitions of companies - allowing moving "exotic" applications into the cloud and out of the datacenter.

## 2.2 Types of E-Learning Systems On Cloud Computing

Cloud Computing is a relatively new term. Amazon's Elastic Compute Cloud (EC2) was the initial offering of a cloud computing platforms that was made available in beta status in 2006. Since then, many solutions have been announced but only a few of them are commercially available as of end of 2010. Nevertheless, Cloud

Computing has a few predecessors - approaches, where compute power has been pooled across the network and made available on demand.

### 2.2.1 Grid Computing

Grid Computing is an approach for pooling compute resources from multiple administrative domains on order to carry for massively parallel computations, such as scientific computing and simulation. Grid computing efforts have been sponsored through research programs of the European Union as well as national research agencies.

Grid computing applications typically follow a "divide-and-conquer" strategy, splitting tasks in (large) sets of subtasks and carrying out those subtasks on multiple (sometimes thousands) of nodes on the grid. Grids may be established and used inside one organization or being spread across organizational boundaries forming so called "Virtual Organizations".

Grid computing is a form of distributed computing forming a "virtual supercomputer". Following the SPMD (Same Program Multiple Data) approach, grid applications often run the same program with varying datasets on multiple nodes in the grid. Those coarsely granular parallel applications can be found in the areas of distributed simulation, climate predication, drug discovery, economic forecast, and the like.

One early example for grid computing efforts is the SETI@HOME project, where users all over the world voluntarily donate idle cycles of their computers to help with analyzing recorded radio data and support the search for extraterrestrial intelligence. Another well-known grid computing project is the World Community Grid - Berkeley Open Infrastructure for Network Computing (BOINC). Using again idle time computing, this effort helps conducting protein-folding experiments in order to create more durable rice crops. Besides those "special purpose" grid initiatives, efforts have been made to establish standard tools and environments to send out jobs to multiple grid nodes and collect results. Most notably is Sun Microsystems’s Grid Engine that forms a cluster computing environment out of standard workstations. The Globus toolkit (GTK) is another de facto standard for setting up grid computing environments. Standardization efforts of the Open Grid Forum (OGF) focus on compilation of WebService standards for usage on the grid.

In contrast to cloud computing, grid computing never considered the interactive end user being part of the scenario. It rather follows a model where jobs are being submitted through portals and results are being retrieved hours later in form of log files and collected program output. Also, grid computing did not solve the problem of integrating administrative domains. To enable interoperability across sites, grid administrators typically have to install certificates, enable ssh (secure shell) login, and setting up mutual trust relationships across all partners. Most importantly, in contrast to providing virtualized compute resources to many users simultaneously (as cloud computing does), grid computing rather focuses on creating a huge, distributed virtual supercomputer, which is managed in a batch-job-processing scheme.

## 2.2.2 Utility Computing

The term utility computing refers to packaging of IT resources (such as CPU, memory, network bandwidth, storage) into a metered service similar to traditional utilities (such as the telephone network). With low initial costs, utility computing relies on a pay-per-use billing model and allows quick reaction to changes in demand of IT services.

Utility computing is not a new concept but has a rather long history in the world of ultra-reliable but expensive mainframe computers. IBM typically sells computing capacity rather than physical processor and storage resources. Virtualization techniques have been in place on the mainframe for many decades. They provide the underpinnings for effectively sharing compute resources across multiple users and organizations, thus establishing the notion of compute resources as a utility.

Client/server computing and the PC marked a departure from the traditional world of mainframe computing. However, with the massive increase in computing capacity in the PC architecture, the establishment of huge, underutilized datacenters, and the advent of virtualization support in Intel's and AMD's CPUs, all the prerequisites for managing computation as a pay-per-use service have been in place for ten years. Cloud computing extends the notion of utility computing by including the client side in the picture.

## 2.2.3 Autonomic Computing

The term autonomic computing initially has been coined by the IBM. It describes the ability of systems to be more self-managed. Self-management is seen as a promising approach to cope with the ever-increasing complexity of computing systems and infrastructures.

In an autonomic system, the human operator does not control the system directly. Instead, he defines general policies and rules that serve as an input for the self-management process. The following four functional areas have been defined for autonomic systems:

- Self-Configuration: Automatic configuration of components;
- Self-Healing: Automatic discovery, and correction of faults;
- Self-Optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements;
- Self-Protection: Proactive identification and protection from arbitrary attacks.

In order to be manageable, cloud computing infrastructures (often called "the fabric") have to follow autonomic computing principles. Moving workloads transparently from one compute node to another (with little or no blackout time) can be seen as a self-healing mechanism in the cloud. The ability to instantiate certain pre-configured machine images (as with Amazon EC2) or certain roles (as with Windows Azure) can be seen as a step towards self-configuration. Replication in space (redundancy) or time (re-execution) for both, computation and data is the typical approach for self-optimization and self-protection found in the cloud.

## 2.2.4 Dynamic Datacenter Alliance

With its Dynamic Data Center Alliance, Microsoft pursues similar goals like the autonomic computing initiative.

Microsoft's Dynamic Data Center Toolkit enables data center managers to optimize resources, improve visibility of IT assets and issues, meet service levels, and increase efficiencies—all while decreasing costs. The System Center Data Center Management solutions provide:

- Configuration management: Automated provisioning and updating with server consolidation through virtualization.

- End-to-end monitoring: Application and service-level monitoring with proactive platform monitoring.
- Server compliance: Configuration controls and reporting plus centralized audit of system security.
- Data protection and recovery: Backup and restore in addition to business continuity through server virtualization.

Most notably with Microsoft's approach is the attempt to form and define an ecosystem of partners including hosting companies, system integrators, hardware manufacturers, and software vendors who are working with Microsoft namely to offer differentiated value to customers based on the Dynamic Data Center Toolkit (for Hosters and Enterprises). However, principles of the dynamic datacenter are applicable to the operation of cloud environments as well. In fact, VMware's vCloud initiative is exactly suggesting an ecosystem of partners (hosters) to build (private) clouds.

## 2.2.5 Hosting / Outsourcing

Just like cloud computing, traditional hosting/outsourcing offerings typically suggest the migration of a range of business applications and related services, such as backup, from a company's data center onto an infrastructure owned, managed, and operated by an external provider. Typically, there are service-level-agreements (SLAs) associated with software and services operated by that provider.

However, there are two major differences between hosting and cloud computing: The deployment of applications on an off-premise hoster's infrastructure has to obey certain rules and practices. These may be time consuming. For test installation, development setups, or rapidly and suddenly changing applications these procedures may be just too inflexible. Cloud computing can shorten certain procedures. There are just fewer partners involved and the developer is basically able to manage his infrastructure on his own. This gives them freedom but imposes also a burden.

The second difference between cloud computing and off-premise hosting is in the potential amount of savings in terms of operational cost, management (administrative) overhead, and even carbon footprint. While many hosters operate applications and services for multiple clients and thus are able to consolidate multiple workloads on fewer physical machines, the potential amount of sharing on the level of the hardware infrastructure is even bigger in the huge datacenters operated by cloud providers.

Cloud computing is a young field - with commercial offerings by some providers available for just over a year. Therefore, it is difficult to judge whether potential savings due to the economy of scale really will be reflected in the overall operational costs of clients. Depending on the size of an enterprise, cloud computing and traditional hosting will just fall in the same order of magnitude with respect to operational costs.

# Chapter 3 : Project Architecture

## 3.1 E-Learning using Cloud Computing

Education or Learning is an important component of life and No human beings are able to survive properly without education. Now a days, there are lots of paradigms for getting knowledge or learn something. One of the most promising paradigms for education is e-learning. E-learning is commonly referred to the intentional use of networked information and communications technology (ICT) in teaching and learning. Some other terms are also used to describe this mode of teaching and learning including online learning, virtual learning, distributed learning, network and web-based learning. The growth of e-learning is directly related to the increasing access to ICT, as well as its decreasing cost. The capacity of ICT to support multimedia resource-based learning and teaching is also relevant to the growing interest in e-learning. Poor or insufficient technology infrastructure can cause more damage than good to teachers, students and the learning experience. While the costs of the hardware and software are falling, often there are other costs that have not been factored into the deployment of e-learning ventures. The most important of these include the costs of infrastructure support and its maintenance and the appropriate training of staff to enable them to make the most of the technology.

Cloud computing applications provide flexibility for all educational universities, schools and institutions. The cloud platform in institutions' campuses provides effective infrastructure and deployment model for their dynamic demands. The benefits of cloud computing can support education institutions to resolve some of the common challenges such as cost reduction, quick and effective communication, security, privacy, flexibility and accessibility. "Cloud computing" is the next accepted action in the evolution of on-demand information technology services and products. Cloud computing allows to move the processing effort from the local devices to the data center facilities. The software is seen as a service and the applications and data are stored on multiple servers that can be accessed from the Internet. However, in traditional web-based e-learning mode, system construction and maintenance are located in interior of educational institutions or enterprises, which results in a lot of problems existed. cloud computing has many advantages such as expected performance, reduced upfront investment (i.e., software, hardware, and professional staff to maintain servers and upgrade software), high availability, reduced launching time, infinite scalability, tremendous fault-tolerance capability, and accessibility, enhanced collaboration, and mobility, allow users to use any device, such as a mobile phone, personal computer (PC) etc. Cloud computing is becoming an attractive technology due to its dynamic scalability and effective usage of the resources; it can be utilized under circumstances where the availability of resources is limited. This paper presents the impact of using cloud computing upon e-learning solutions development.

E-learning includes all forms of electronically supported learning and teaching. The information and communication systems, whether networked learning or not, serve as specific media to implement the learning process. This often involves both out-of-classroom and in-classroom educational experiences via technology, even as advances continue in regard to devices and curriculum. Abbreviations like CBT (Computer-Based Training), IBT (Internet-Based Training) or WBT (Web-Based Training) have been used as synonyms to e-learning.

## 3.2 Tools and Technologies Used

### 3.2.1 HTML

HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible 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 to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

### 3.2.2 CSS Coding

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998), and they also operate a free CSS validation service

### 3.3.3 Java

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.[10][11] Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) 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 it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

### 3.3.4 Java platform

Main articles: Java (software platform) and Java virtual machine.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any 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 they 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.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using bytecode is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time (JIT) compilers were introduced from an early stage that compilebytecodes to machine code during runtime

## 3.3 Cloud Platforms

### 3.3.1 Amazon Web Services - IaaS

Since August 2006, Amazon Web Services (AWS) has provided an infrastructure web services platform in the cloud. By providing a suite of elastic IT infrastructure services, AWS enables access to compute power, storage, and other services. AWS operates on a pay-per-use basis, with no up-front expenses or long-term commitments. This makes AWS a cost-effective way to deliver applications to customers and clients. AWS operates on Amazon.com's global computing infrastructure. The term "cloud computing" initially has been coined for a few Amazon Web Services, namely the Elastic Compute Cloud (EC2) and the Simple Storage Service (S3).

Amazon Web Services is more than a collection of infrastructure services. It incorporates identity, payment, database, messaging, and other services. All AWS services are priced on a pay as you go model, with no up front expenses or long-term commitments.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. EC2 can be categorized as an Infrastructure-as-a-Service (IaaS) offering. It allows configuring an Amazon Machine Instance (AMI) and loading it into the Amazon EC2 service. By running multiple AMIs simultaneously, EC2 allows to quickly scale capacity, both up and down, as computing requirements change.

Amazon Simple Storage Service (Amazon S3) is a simple web services interface that can be used to store and retrieve large amounts of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. Data on S3 is unstructured (blobs). There exists no interface to run structured queries or operations on the data on S3 directly from the client. Instead, SimpleDB adds an additional layer on S3 to provide such an interface.

With Amazon CloudFront, content delivery on the Web can be implemented. Even with thin clients, cloud services need a certain portion of user interface running on the client's desktop in order to be usable. CloudFront is the proposed solution for hosting those downloadable user interfaces - in the form of a collection of JavaScript/Ajax code. Cloud Front integrates with other Amazon Web Services.

Amazon SimpleDB is a web service for running queries on structured data in real time. This service works in close conjunction with Amazon Simple Storage Service (Amazon S3) and Amazon Elastic Compute Cloud (Amazon EC2), collectively providing the ability to store, process and query data sets in the cloud. Amazon SimpleDB provides the core functionality of a database—real-time lookup and simple querying of structured data.

Amazon Simple Queue Service (Amazon SQS™) is a reliable, highly scalable, hosted queue for storing messages as they travel between computers. By using Amazon SQS, developers can simply move data between distributed components of their applications that perform different tasks, without losing messages or requiring each component to be always available. Configuration rules for EC2 can be based on SQS attributes and dynamically start up or shutdown AMI instances as queue length vary. This allows for implementation of self-adaptive behavior in cloud services.

With EC2, S3, and SQS, Amazon has provided the first implementation for a public cloud. Public clouds are appealing for many use cases, where data security, privacy, and ownership are not of primary concern. This is true for use cases like scientific computing, simulation, image processing, video rendering, and even gaming. However, most enterprises view data as their primary asset. Private clouds - which are very much comparable to traditional hosting solutions (as discusses above), may be a promising approach to keep important data absolutely private.

Amazon Virtual Private Cloud (Amazon VPC) is an extension of the initial, purely public cloud. It provides a secure and seamless bridge between a company's existing IT infrastructure and the AWS cloud. Amazon VPC enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities such as security services, firewalls, and intrusion detection systems to include their AWS resources. Amazon VPC integrates today (beta status) with Amazon EC2, and will integrate with other AWS services in the future.

When developing for the Amazon cloud, one has to distinguish between APIs and tool support for deploying, managing and monitoring instances of Amazon Machine Images (AMIs) in the cloud and between APIs and tools for developing applications running on AMIs in the cloud. Being an IaaS solution, AWS tools mainly focus on managing AMIs for EC2 and setting up SimpleDB configurations for S3. Based on the Eclipse Web Tools Platform, the AWS Toolkit for Eclipse guides Java developers through common workflows and helps to configure Tomcat servers, run applications on Amazon EC2, and debug the software remotely through the Eclipse IDE. There exists also a .NET library that wraps AWS APIs and allows management of AMI instances (preferably running Windows and IIS) from .NET client programs. Similar wrapper libraries exist for Perl, PHP, Ruby and Python.

In the remainder of the section we will take a closer look on EC2, S3, and SQS.

### 3.3.2 Google AppEngine - PaaS

Google App Engine allows running web applications on Google's infrastructure. This effectively allows clients to build their own Software-as-a-Service applications. App Engine includes the following features:

- Dynamic web serving, with full support for common web technologies
- Persistent storage with queries, sorting and transactions
- Automatic scaling and load balancing
- APIs for authenticating users and sending email using Google Accounts
- A fully featured local development environment that simulates Google App Engine on a client computer
- Task queues for performing work outside of the scope of a web request
- Scheduled tasks for triggering events at specified times and regular intervals

Applications can run in one of two runtime environments: the Java environment, and the Python environment. Each environment provides standard protocols and common technologies for web application development. Applications can be made accessible either from a client's domain name (using Google Apps) or through Google's reserved domain called appspot.com. Applications can be shared with the world, or within an organization. App Engine supports integrating an application with Google Accounts for user authentication.

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, you can build your app using standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the Python standard library. The Java and Python runtime environments are built to ensure that applications run quickly, securely, and without interference from other apps on the system.

Applications run in a secure environment that provides limited access to the underlying operating system. These limitations allow App Engine to distribute web requests for the application across multiple servers, and start and stop servers to meet traffic demands. The sandbox isolates an application in its own secure, reliable environment that is independent of the hardware, operating system and physical location of the web server.

Examples of the limitations of the secure sandbox environment include:

An application can only access other computers on the Internet through the provided URL fetch and email services. Other computers can only connect to the application by making HTTP (or HTTPS) requests on the standard ports.

An application cannot write to the file system. An app can read files, but only files uploaded with the application code. The app must use the App Engine datastore, memcache or other services for all data that persists between requests.

Application code only runs in response to a web request, a queued task, or a scheduled task, and must return response data within 30 seconds.. A request handler cannot spawn a sub-process or execute code after the response has been sent.

App Engine costs nothing to get started. All applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month, absolutely free. When you enable billing for your application, your free limits are raised, and you only pay for resources you use above the free levels.

### 3.3.3 Salesforce.com - SaaS / PaaS

Salesforce.com was founded in 1999 as a company specializing in Software-as-a-Service (SaaS) offering for Customer Relationship Management (CRM). It was really one of the pioneers of the SaaS model of distributing software on the Internet.

Salesforce.com promises availability of 99.9% for its cloud infrastructure. This figure is comparable with the service levels reached by the competitors. However, the trust.salesforce.com site is truly unique. It shows both recent and historical performance plus uptime data for the Salesforce.com cloud infrastructure.

Salesforce.com's CRM solution is broken down into several modules: Sales, Service & Support, Partner Relationship Management, Marketing, Content, Ideas and Analytics. In order to ease customization and allow for extension of the CRM SaaS solution, Salesforce.com has built up AppExchange. Launched in 2005, AppExchange is a directory of applications built for Salesforce by third-party developers, which users can purchase and add to their Salesforce environment. As of September 2008, there are over 800 applications available from over 450 ISVs. Applications include services from Google, Constant Contact, Vertical Response, Good Data, and Box.net. The Force.com platform is a further generalization of the initial SaaS solution. Salesforce users can customize their CRM application on the platform- or the tab-level. In the system, there are tabs such as "Contacts", "Reports", and "Accounts". Each tab contains associated information. For example, "Contacts" has standard fields like First Name, Last Name, and Email. Customization can be done on each tab, by adding user-defined custom fields. Customization can also be done at the "platform" level by adding customized applications to a Salesforce.com instance; that is adding sets of customized / novel tabs for specific vertical- or function-level (Finance, Human Resources, etc) features.

In addition to the web interface, Salesforce offers a Web Services API that enables integration with other systems and has wrapper libraries for programming languages such as Java, VB.NET, C#, and other .NET languages.  In April 2009, Salesforce released a slimmed down version of their application for subscribers with Blackberry, iPhone, and Windows mobile devices.

## 3.4 Comparison of Cloud Platforms

In the previous sections we have discussed eight different approaches to cloud computing, which can be seen as representatives for a growing group of offerings. We have presented Infrastructure-as-a-Service (IaaS) offerings, namely Amazon Web Services (AWS) and VMwaresvCloud. We have discussed three Platform-as-a-Service (PaaS) solutions, namely Google AppEngine, Microsoft's Windows Azure, and Salesforce.com's force.com. Finally, we have given an overview over three Software-as-a-Service (SaaS) solutions, namely Salesforce.com, Microsoft's OfficeLive, and Google Apps.

From a software architect's and developer's standpoint most interesting are the IaaS and PaaS solutions. Those solutions allow for architecting your own applications that may be hosted on the cloud. In contrast, SaaS offerings typically only lend themselves to some limited configuration.

### 3.4.1 Common Building Blocks

All cloud computing platforms build on a similar set of building blocks. These are:

- Virtualization: In order to dynamically increase and decrease computing capacity in response to changing workloads or administrative triggers, physical computers have to be abstracted away. Virtualization of compute resources can be achieved on many different layers of abstraction (application server, operating system processes, virtual machines, logical partitions of physical hardware). For cloud computing environments, the logical machine level of abstraction has proven to work well. Virtual machines can easily be replicated when load increases. They can be moved across physical machines for consolidation purposes or to allow for system maintenance.
- WebServices: All cloud-computing platforms allow for access to cloud resources via WebService APIs. These APIs typically are hidden behind wrapper libraries in the common programming languages, such as Java, C#, C++ (and other .NET languages) for the client as well as "web languages" such as Python, Ruby, and PHP.
- Orchestration / Service Bus: All cloud-computing platforms provide some means for integrating services in the cloud. These services include compute services, data stores, and messaging middleware. Platforms differ with respect to their support for integrating external services (hosted on premise or at partner sites) with cloud services.
- Clients-side user interfaces: Cloud computing applications need to expose a client-side user interface. Thus, all cloud-computing platforms provide some means for interacting with the user. This may be as simple as a portal that integrates mashups shown in a web browser. Often, user interfaces can be based on Ajax and JavaScript. However, the most elaborate platforms rely on a fully fledged component model such as JavaBeans/Applets or Silverlight/.NET for programming user interfaces that are downloadable and can be executed on the client's computer.

### 3.4.2 Which Cloud to choose ?

Software-as-a-Service (SaaS) solutions have been the initiators of the cloud computing excitement. There are far more SaaS solutions available than the ones we have discussed here. From an end user's perspective, there are a few questions that need to be answered when it comes to using SaaS systems. The most important ones are: "Who owns the data?" and "Can I work offline?" Systems like Salesforce.com CRM and Google Apps ask the user to fully trust the provider for keeping data secure, consistent, and available. Both services are of no use when the computer is not connected to the Internet. On the other hand, they provide unprecedented ease-of-use without any need to install and maintain software on the client's computer.

Microsoft's OfficeLive follows a different route. While still allowing (read) access to the data from anywhere on the Internet, it will download the data to a client's computer in order to implement modify or write operations. This puts a small burden on the user - as he needs to maintain some software on his computer. However, it provides the big advantage of having the data accessible even when the computer is not connected. Also, it is far easier to keep backups of data if local copies are maintained anyway. Another aspect is the richness of the interface: Software on a local computer today still provides for a better user experience than Web 2.0 interfaces.

Amazon's Elastic Compute Cloud (EC2) was the first cloud solution on the market that was appealing not only to the administrator and end-user but also to developers. Infrastructure-as-a-Service (IaaS) historically presented the second step in cloud computing. Amazon Web Services (AWS) define the de facto standard in IaaS. They have defined all the building blocks, like the concept of an Amazon Machine Image and the use of virtual machines, the Simple Storage Service, and the message queuing middleware. AWS also has a long-standing record in providing high levels of service availability. The AMI concept is very flexible - developers can basically pre-configure their virtual machine with all necessary tools, such as application servers, databases, and middleware solutions prior to deploying them on the cloud (via S3). The biggest drawback of the concept is the burden of maintenance - which is still on the developer, as they have to keep the software stack on their AMI in an updated and well-maintained fashion. Amazon is certainly working on this issue and tries to ease this burden by providing certain pre-configured machine images. One may argue that the AWS cloud will slowly move from an IaaS offering to a PaaS offering.

VMware's vCloud, the other IaaS offering discussed here is the most recent addition to the cloud-computing spectrum. In contrast to all other cloud platforms discussed here, VMware envisions a whole cloud-computing ecosystem where independent software vendors and hosters may provide competitive offers for running virtual data centers in the cloud. Having multiple providers may circumvent the risk of vendor lock in and also mitigates the business risk of being dependent on just a single cloud operator. VMware tries to standardize on the interfaces necessary to configure, load, instantiate, move, and replicate virtual machine instances in the cloud. It is suggesting using the Open Virtualization Format (OVF) as the standard for cloud computing instances. VMware has a long-standing record on virtualization. Playing just on the OVF level will allow VMware to build upon its strengths and implement monitoring and system administration interfaces for cloud instances on a generic level. Improved monitoring and self-adaptive management capabilities would be a most desirable feature for all the cloud platforms currently available.

Platform-as-a-Service (PaaS) solutions are - from an architect's and developer's standpoint - the wave of the future. The integration of elaborate development tools on the client side with a standardized and well-maintained execution environments in the cloud provides for ease-of-use and high efficiency. PaaS platforms differ in the set of supported programming languages and the set of additional infrastructure services available with the execution environment in the cloud.

Salesforce.com's Force.com PaaS offering is tightly integrated with a database in the cloud. It is best suited for a specialized set of cloud applications, where customer data is in the center of operation. However, developers have to use the proprietary Apex language in order to code for the cloud. Force.com applications therefore are hardly portable. There is little support for running cloud applications on-premise.

Microsoft's Windows Azure and Google's AppEngine are quite comparable in their basic approach to cloud computing. Both are PaaS offerings that promise ease-of-use and simple maintenance of cloud applications to the architect and developer. Both platforms support integration between powerful client-side integrated development environments - namely Visual Studio for Windows Azure and Eclipse for AppEngine

- and the cloud. Both providers have a long-standing trust record for providing services on the Internet.

Microsoft supports the usage of all .NET languages (C#, VB.NET, C++, and many more) with Windows Azure. Google provides a tight integration of Java (and languages supported by the JVM) with AppEngine. Both platforms allow for installation of additional software and there are interoperability stories explaining how to run code written in other languages on Windows Azure and AppEngine.

The biggest difference between AppEngine and Windows Azure is the integration of SQL Azure in Microsoft's cloud computing platform. The availability of a database in the cloud that supports structured queries and complex operations on data may be important for a wide range of applications. Another specialty of Windows Azure is the Service Bus integrated in AppFabric. Not only does the service bus support orchestration of services in the cloud, it also allows reaching out from the cloud onto services running locally in a client's or partner's datacenter. Obviously there is a tradeoff: accessing off-cloud services allows for easier system integration on one hand side, it makes service availability directly dependent on those external services on the other hand side.

## 3.5 Cost and Productivity Analysis

### 3.5.1 Improved training and material costs

With e-learning, each time the course is accessed our return on investment improves because users are dividing the fixed production costs by number of uses. We also have savings through decreased travel, reduced material, and hopefully improved (and more efficient) performance.

### 3.5.2 Increased productivity

Because e-learning is not bound by geography or time, you can control training's impact on production by training people during down times. In addition, with the current economy, you're asking people to do more with less. So e-learning is a great way to give them the tools and skills needed to enhance their performance.

# Chapter 4 :Project Design

## 4.1 System Development Life Cycle

The Software development life cycle ,also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

The six-step process contains a procedural checklist and the systematic progression required to evolve an IT system from conception to disposition. The following descriptions briefly explain each of the seven phases of the SDLC:

### 4.1.1 Conceptual Planning.

This phase is the first step of any system's life cycle. It is during this phase that a need to acquire or significantly enhance a system is identified, its feasibility and costs are assessed, and the risks and various project-planning approaches are defined. Roles and responsibilities for the Asset Manager, Sponsor's Representative, System Development Agent (SDA), System Support Agent (SSA), and other parties in SDLC policy are designated during this stage and updated throughout the system's life cycle.

#### 4.1.1.1 Planning and Requirements Definition.

This phase begins after the project has been defined and appropriate resources have been committed. The first portion of this phase involves collecting, defining and validating functional, support and training requirements. The second part is developing initial life cycle management plans, including project planning, project management, Configuration Management (CM), support, operations, and training management.

**4.1.1.2**. During this phase, functional, support and training requirements are translated into preliminary and detailed designs. Decisions are made to address how the system will meet functional requirements. A preliminary (general) system design, emphasizing the functional features of the system, is produced as a high-level guide. Then a final (detailed) system design is produced that expands the design by specifying all the technical detail needed to develop the system.

1. **Implementation.:**

   During this phase, the new or enhanced system is installed in the production environment, users are trained, data is converted (as needed), the system is turned over to the sponsor, and business processes are evaluated. This phase includes efforts required to implement, resolve system problems identified during the implementation process, and plan for sustainment.

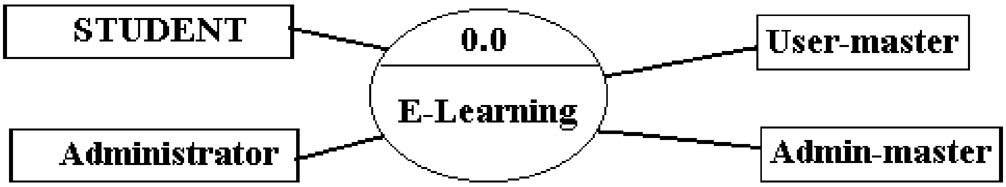
2. **Operations and Maintenance:**

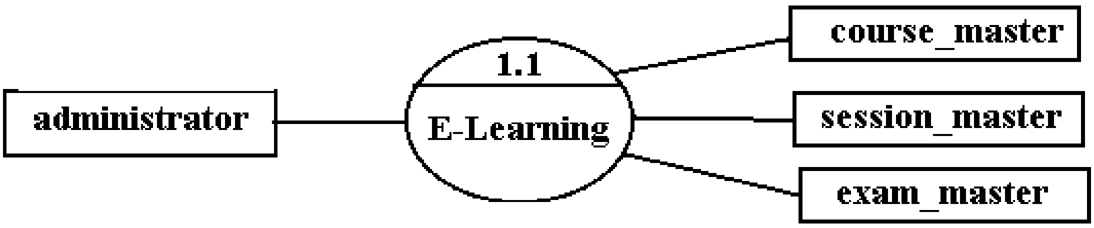
   The system becomes operational during this phase. The emphasis during this phase is to ensure that sponsor needs continue to be met and that the system continues to perform according to specifications. Routine hardware and software maintenance and upgrades are performed to ensure effective system operations. User training continues during this phase, as needed, to acquaint new users to the system or to introduce new features to current users. Additional user support is provided, as an ongoing activity, to help resolve reported problems.

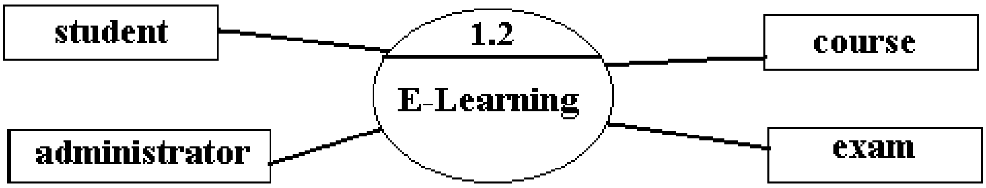
## 4.2CONTEXT LEVEL DFD

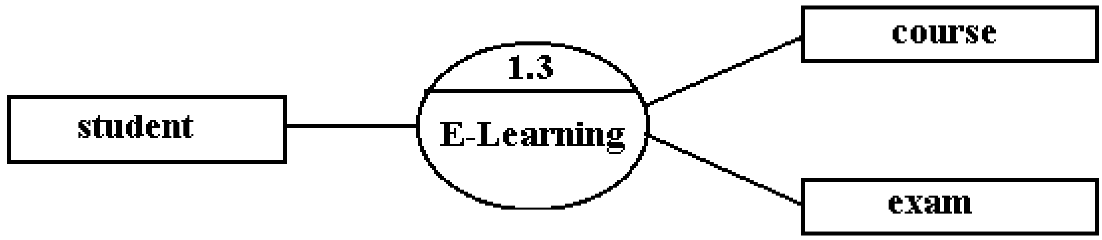
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated.[2] DFDs can also be used for the visualization of data processing (structured design).

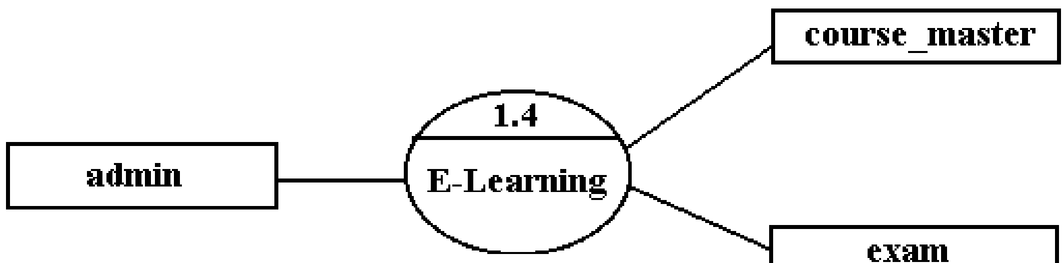
A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.






## 4.3 E-R Diagram

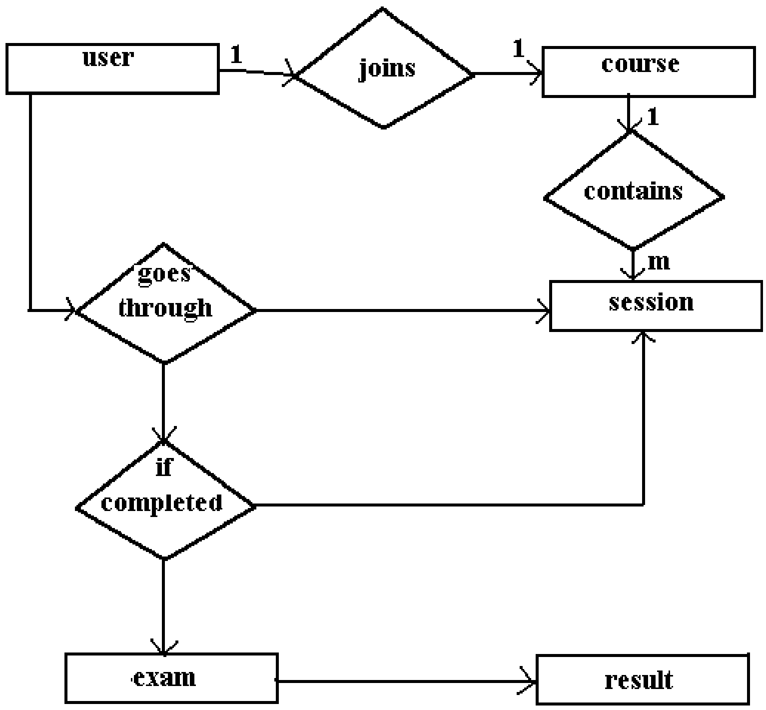
An ER model is an abstract way of describing a database. In the case of a relational database, which stores data in tables, some of the data in these tables point to data in other tables - for instance, your entry in the database could point to several entries for each of the phone numbers that are yours. The ER model would say that you are an entity, and each phone number is an entity, and the relationship between you and the phone numbers is 'has a phone number

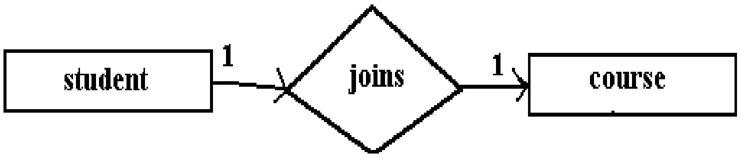
An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. An entity is an abstraction from the complexities of a domain. When we speak of an entity, we normally speak of some aspect of the real world which can be distinguished from other aspects of the real world

Entities and relationships can both have attributes. Examples: an employee entity might have a Social Security Number (SSN) attribute; the proved relationship may have a date attribute.

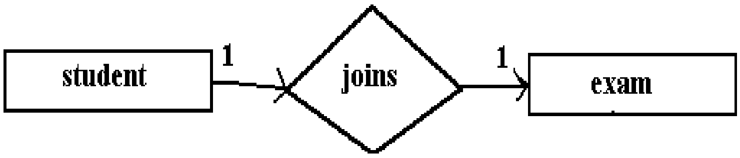
Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

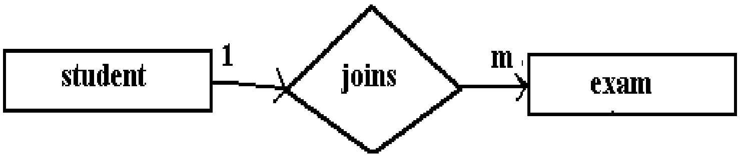
Entity–relationship diagrams don't show single entities or single instances of relations.

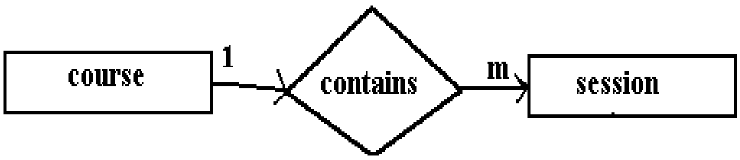



student
1
joins
1
course

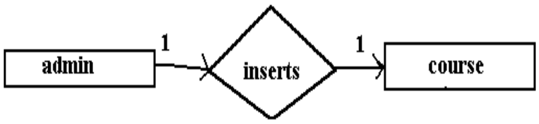

student
1
joins
m
course

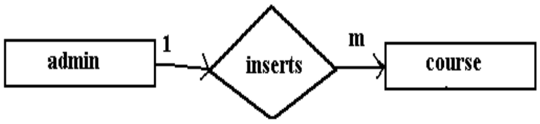

student
1
joins
1
exam


student
1
joins
m
exam


course
1
contains
m
session

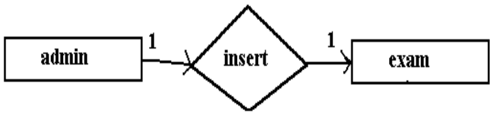

course
1
contains
1
session

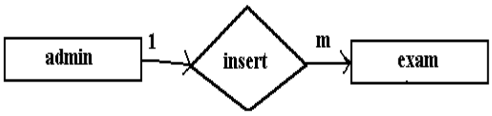

admin
1
inserts
1
course


admin
1
inserts
m
course

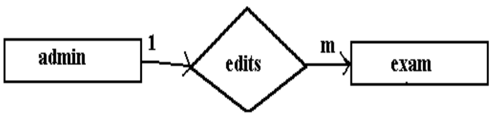

admin
1
edits
m
course


admin
1
edits
1
course


admin
1
insert
1
exam


admin
1
insert
m
exam

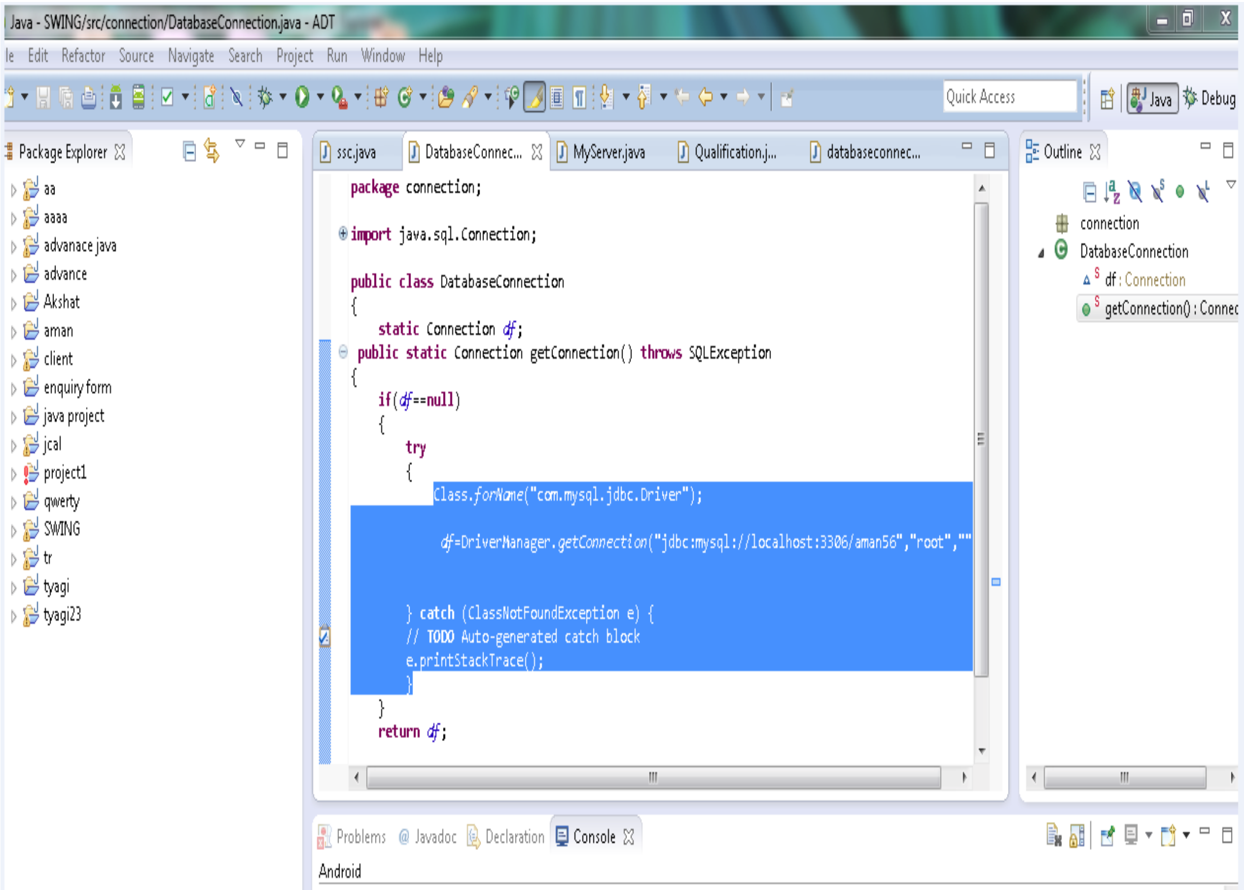

admin
1
edits
1
exam


admin
1
edits
m
exam


student
1
gets
1
result

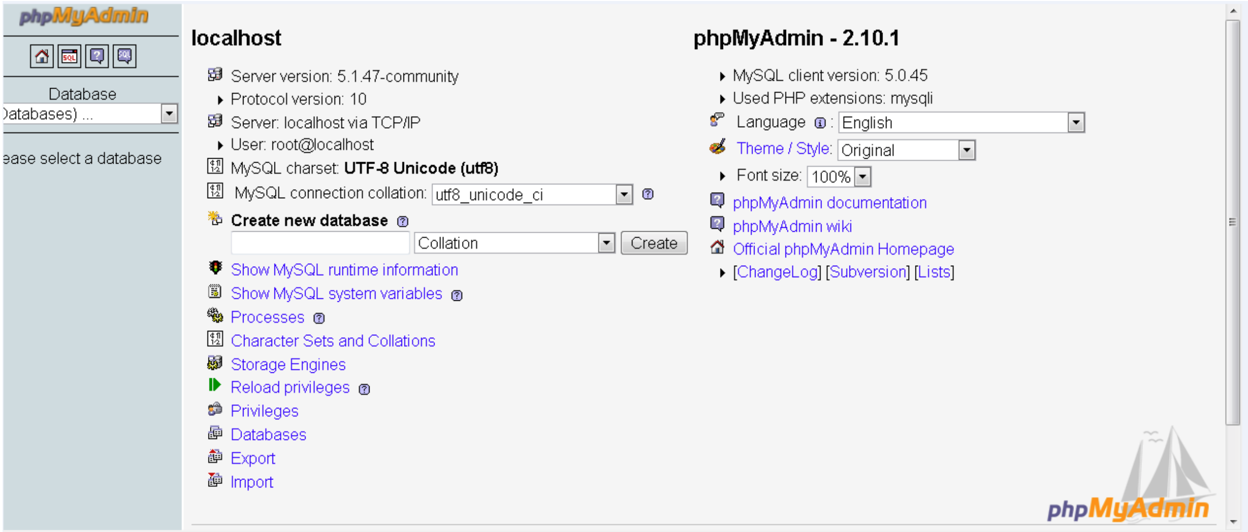
## 4.4 Project Snapshots

### Coding

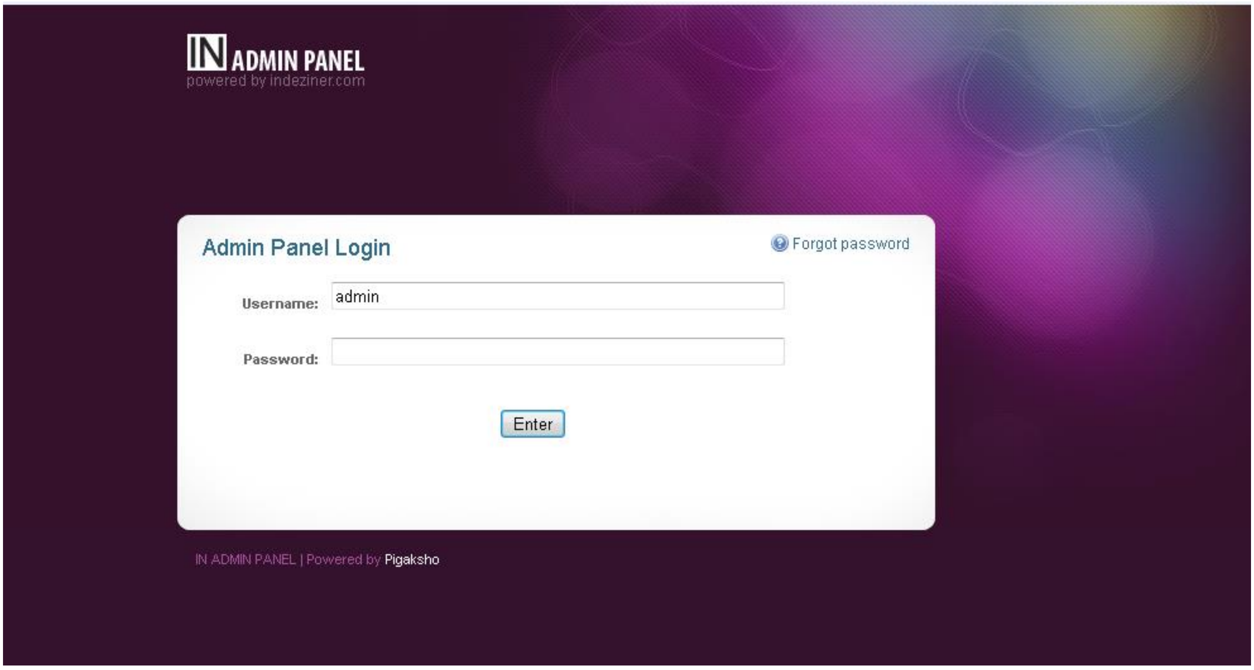



### Wamp Server Source File

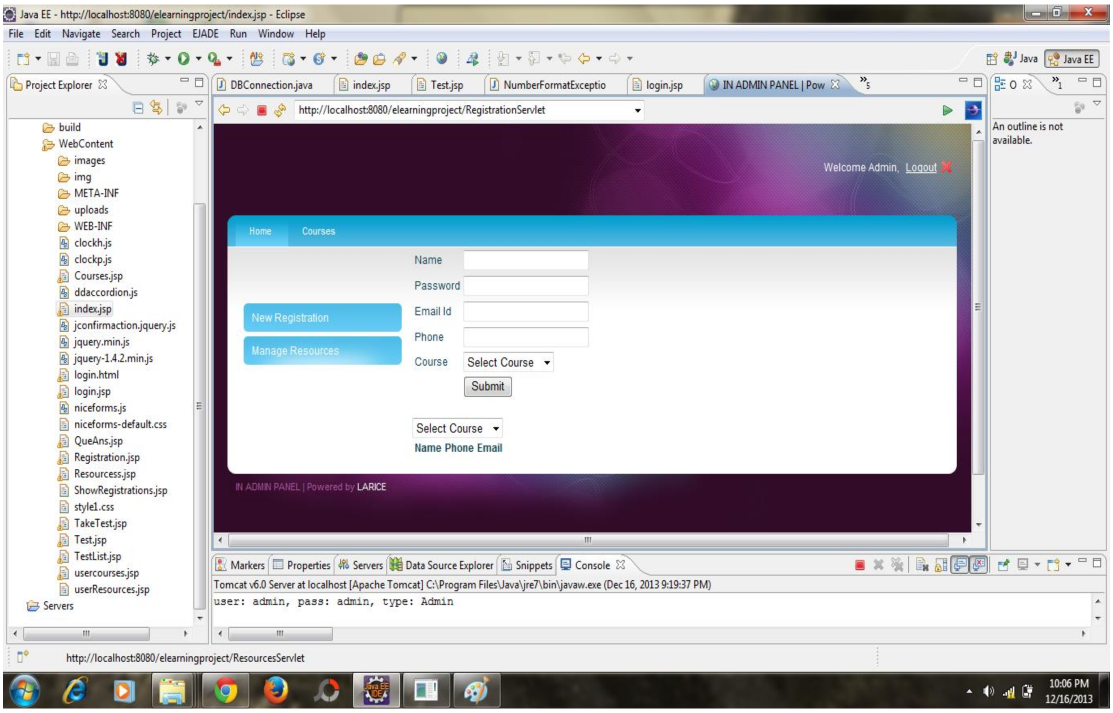
# SQL DATABASE OF PROJECT



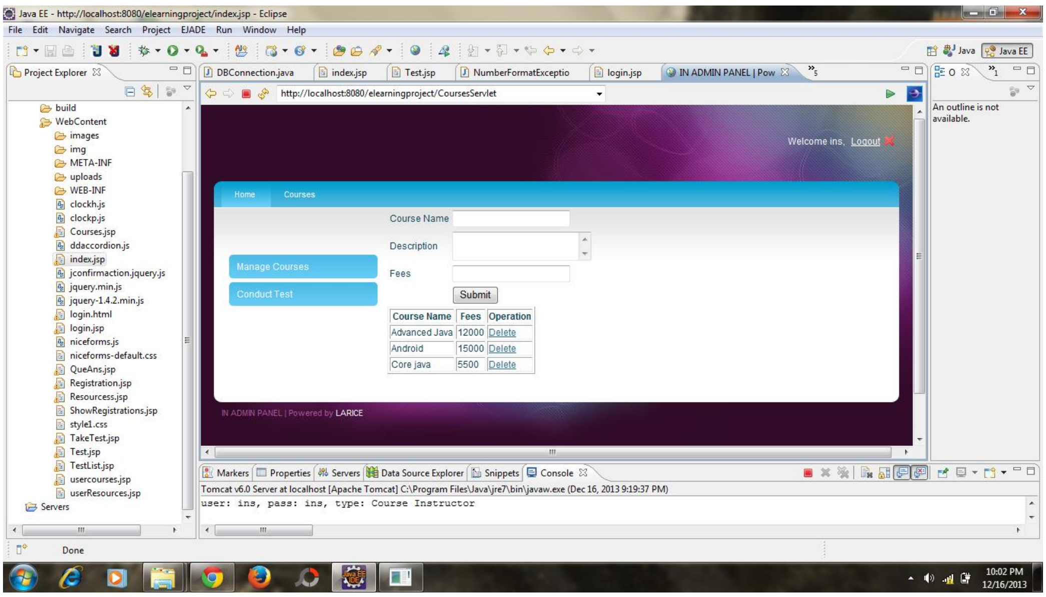

# ADMIN LOGON PAGE



# NEW ENTRY




# ADD NEW COURSE



# Chapter 5: Conclusion

This project has covered the most prominent cloud computing solutions available today. The discussion has focused on technical characteristics and tried to differentiate the cloud computing platforms. It has not evaluated business perspectives, potential savings, or economic opportunities opened up by cloud computing. Given the tremendous amount of experience and know how in system optimization and monitoring present in today's datacenters, another important aspect for successful operation of cloud applications will be the question of how to monitor and manage all layers of the cloud stack. Self-adaptive management mechanisms present in the cloud fabric are only in a very limited way exposed to cloud applications and clients. A consistent monitoring model covering all layers of the cloud stack is badly needed. Even more important is the question of how to secure data in the cloud. Current levels of access control available in the cloud platforms are very limited. Identity management and federation as well as management of trust among clients and services in the cloud will be most important.

E-learning systems provide processes of delivering the learning contents to learners who have different backgrounds, interests, and locations away from a classroom in order to maximize the effectiveness of learning. Usually, the classical e-learning system is based on client/server architecture thus they lack of the scalability, flexibility and interoperability. It makes the learning resources cannot share, and the system improvement is not easily. This paper proposes the cloud computing architecture in the e-learning system that the architecture separate into three layers includes infrastructure, platform and application. This architecture needs to design components in order to transfer the learning resources to the cloud platform. Infrastructure layer, the learning resources from the traditional system are transferred to the cloud database instead of the usual DBMS. Platform layer, a new e-learning system that consists of the CMS, AMS, and other service components were developed. These components were developed to be the intermediary between cloud database and the applications.

# Future Scope

The E-Learning market covers the academics, corporate and consumer fields, and has a variety of segments, including content providers, technology vendors, and service providers. Morgen, Keegar & Co. estimate that expenditure on all forms of education now exceed $750 billion in the USA and $2 trillion world-wide, with revenue growth for e-learning expected to outstrip that in all other sectors of the education industry. The following things e-learning provides in the Future:

- Improving the quality of learning.
- Improving access to education and training.
- Reducing the cost of education.
- Improving the cost-effectiveness of education

# Chapter 7 References


- "**An E-learning Ecosystem Based on Cloud Computing Infrastructure**" – Department of Computer Science, Union University, Jackson, TN, 38305, USA

- "**The NIST Definition of Cloud Computing**" – National institute of Standards and Technology, U.S department of Commerce

- "**Cloud computing: state-of-the-art and research challenges**" Qi Zhang , Lu Cheng , RaoufBoutaba , University of Waterloo, Waterloo, Ontario, Canada

- "**ASPECTS OF E-LEARNING ENVIRONMENTS**" - Thomas Dietinger

- "**Cloud Computing : A Practical Approach**" – Anthony T. Velte, Toby J. Velte, Robert Elsenpeter (Tata McGraw Hill Edition)

- "**Cloud Application Architectures: Building Applications and Infrastructure in the Cloud**" -George Reese

- "**E-learning based on Cloud Computing**", DeepanshuMadan, Computer science &Engg.Deptt. , Dehradun institute of technology Dehradun