Virtual University System

A Case Study of IUT

By:

- Hamisi Ramadhan Mubarak 094304
- Muhammad Ismail 094303
- Yousef Ali Algabal 094301

Supervised By:

Mr. Mahmud Hasan

Assistant Professor Department of Computer Science and Engineering

Computer Science and Engineering

Islamic University of Technology

September, 2012

ACKNOWLEDGEMENT

There is absolutely no doubt that had we not been candidates of sound mind and with good health, we would never have accomplished this demanding task. We, therefore, thank the al-mighty Allah for granting us such success perquisites that cannot be granted by none other than him.

We are indeed very grateful to our distinguished and intuitive supervisor, Mr. Mahmud Hasan, Assistant Professor Department of Computer Science and engineering IUT. He gave us time from his busy schedule and guided us on the right path.

We would also like to thank our Web Architecture teacher Hasan Mahmud who did a remarkable job for introducing J2EE and professional web architectures.

DEDICATION

To my Mom

- Muhammad Ismail

I dedicate this to my family whom, encouragement and motivation made me capable to accomplish this job. Who are always there to support me.

- Yousef Ali Algabal

I dedicate this to my family and friends who have given me support despite being a thousands of miles away from them.

- Hamisi Ramadhan Mubarak

ABSTRACT

The advent of Internet has brought a whole new array of application in our lives. It is changing the way we live, communicate, buy or sell, seek knowledge and entertain. Today we see hundreds of universities across the globe providing education in various fields with people attending real, physical classes. We take this very idea of higher education and gear it by the power of Internet to convey distance education without putting limitation of time and place on people. We call this internet based system of education Virtual University System. People around the globe, who satisfy the basic requirements, can apply for admission in different graduation programs, take virtual lectures, do real assignments, take quizzes and sit for real examination. On the other side of the picture, teacher spent real time to prepare lectures, assess students and prepare examination questions. Though virtual university system is not a replacement of the traditional brick-and-mortar universities, it is just a variant of the former using different set of tools and approach to education.

Chapter 1 Introduction

This document is a specification of final semester project. The title of the said project is Virtual University System (in this case virtual IUT). It gives all the ins and outs of development process from the first step to its final approval presentation. The purpose of this document is to present detailed information to the higher ups (in this case supervisor and teachers), maintenance department and other interested individuals.

1.1 Background

Virtual University System (VUS) is a form of distance education. Students from remote places around the world can register in different disciplines of studies and after successful completion of their academic activities, they are given respective degrees. This system of education is totally based on the Internet but at same time people are doing physical work and studies to get through different semester. In addition to anytime, anywhere access, the candidates have also access to a lot other multimedia tools which enhance their capabilities of learning and improve their qualities of exploration.

VUS in addition to multimedia rich contents provide the web 2.0 standard tools like instant messaging, e-mailing, blogging, discussion forums, bookmarking etc. which acts like a catalyst in the process of knowledge acquisition. Keeping all these benefits in view, we can confidently say that VUS though is not a replacement of traditional system of education but is a versatile system of its own unique nature.

1.2 Motivation

Our attention was drawn to this corner of education by the following three powerful rationales which gave us force to study, analyze and try to develop a model system which imitate this very idea.

1.2.1 Geographical Factor

Higher education has been a big challenge for countries where much of the population live in villages and remote-cum-backward areas. Though most of the countries are trying to establish remote campuses of main universities to render higher education but they are not omnipresent. To fill up the gap, there is a need for an alternative system for which people do not have to travel long distances but is available at their doorsills. And this is what we call Virtual University System.

1.2.2 Busy Life

We know in today's life people are busy with their regular job or business or some other kind of routine work. They find it hard to dedicate time for going to college or university to attend regular

classes. So this is where a Virtual University System comes to rescue, which help them to do their academic stuff anywhere, anytime.

1.2.3 Expanse of Internet

Vast majority of world's population today have access to Internet as compared to a few decades ago. Any service which is based on Internet can be availed by people from anywhere in the world. This omnipresence and widespread usage of Internet are the promising forces to the success of our system. Taking care of the different parameters of the education, Internet is almost the cheapest tool for educating people remotely.

1.3 Problem Identification

The challenge is to build such a system which is providing education to people without forcing them to be physically present in the class or bound by the dimension of time while keeping the standard of education equivalent or competitive to regular system of education. The answer to this is a system just like traditional universities but offering its services through Internet or virtually. More elaborately, the system must offer degrees, certifications and diplomas in different types of courses like short courses, bachelor degrees and master degrees. Each program will comprise of semesters depending on its duration for example for a four year degree program there will be eight semester similar to traditional university system. When we talk about education several questions pop up in our mind like

- Registration of students
- What are the prerequisites of a specific course?
- Will there be entry test or not?
- Age limits
- How many students can be accommodated in a course?
- Number of semester
- In what format the course contents should be available to student for better understanding?
- Submitting assignments and evaluation gradation scheme
-

1.4 Objectives

As every human action have some objectives on the way to achieve. So is our effort not far from clearly defined objectives. The following list is its reflection.

- To provide remote education.
- To remove the boundaries of time and location from higher education.
- To make available the opportunity of cheap education to everyone.
- To simplify the process of seeking and getting admission.
- To use the ubiquitous interface of the web, to which everybody is familiar, to handle academic activities like assignments, distribution of lectures, notifications and news, schedules, evaluation, result delivery etc.

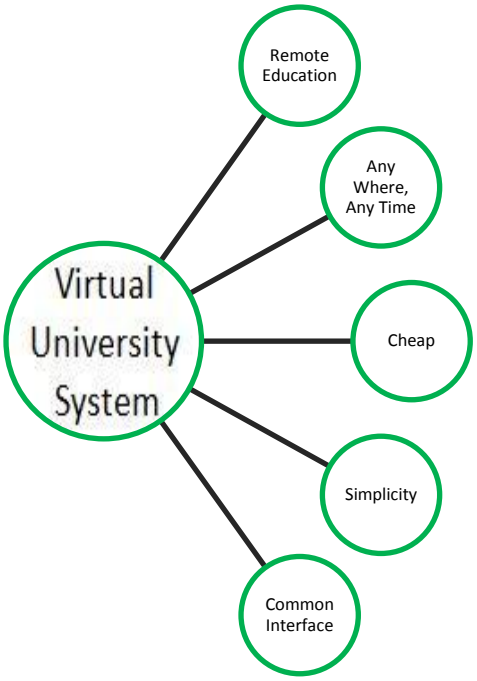

Figure 1.1

1.5 Scope

When we talk about scope we mean how wide the application area of the object of our concern is. Though our idea is so universal and such a powerful that one can apply it anywhere and to any society. It can be narrowed down to serve a specific country while implementing the education policies specific to that country. But we go one step forward in terms of specification by making it a case study considering Islamic University of Technology Bangladesh as a model. The reason why we are using IUT as a model is that that we, the developers are the students of this university and we know everything about it. To make it more clear it provides us with a clear vision how the courses are handled. How many quizzes, assignments or other academic stuff would be there for each individual course and what are the departments, what programs they are offering and are their duration and examination system. In a nutshell, we are equipped with all information that we need for a solid and workable system to build. As virtual university systems are just like traditional system but with an exception that almost nighty percent of the activities are handled through Internet.

Chapter 2 Literature Review

2.1 History

The defining characteristic of all forms and generations of distance education is the separation of student and teacher in time and space. Distance education can be seen as the precursor to online learning. Before the advent of virtual universities, many higher education institutions offered some distance education through print-based correspondence courses. These courses were often referred to as a “course in a box”.These have been developed so that students can obtain almost immediate feedback from professors and online tutors through e-mails or online discussions.

When the term “virtual” was first coined in the computational sense, it applied to things that were simulated by the computer, like virtual memory. Over time, the adjective has been applied to things that physically exist and are created or carried on by means of computers.

The Open University in the United Kingdom was the world’s first successful distance teaching university. It was founded in the 1960s on the belief that communications technology could bring high quality degree-level learning to people who had not had the opportunity to attend campus universities.The idea for a “wireless university” was first discussed at the BBC (British Broadcasting Corporation) by the educationalist and historian J.C. Stobbart. From these early beginnings more ideas came forth until finally the Labour Party under the leadership of Harold Wilson formed an advisory committee to establish an Open University.

With the goal of bringing higher education to all those who wanted to access it, the committee came up with various scenarios before settling on the name Open University. The first idea floated in the UK was to have a “teleuniversity” which would combine broadcast lectures with correspondence texts and visits to conventional universities. In the “teleuniversity” scenario courses are taught on the radio and television and in fact many universities adopted the use of this technology for their distance education courses. The name “teleuniversity” morphed into the "University of Air” which still had the same goal of reaching the lower income groups who did not have access to higher education. The name “University of Air” did not stick and by the time the first students were admitted in January 1971 the name had become what it is today “Open University”. OU proved that it was possible to teach university-level courses to students at a distance.

By 1980, total student numbers at OU had reached 70,000 and some 6,000 people were graduating each year. The 1980s saw increased expansion continue as more courses and subject areas were introduced; as the importance of career development grew, so the university began to offer professional training courses alongside its academic programmes. By the mid-nineties the OU was using the Internet. As of 2008, more than 180,000 students were interacting with OU online from home.

The idea of a virtual university as an institution that used computers and telecommunications instead of buildings and transport to bring students and teachers together for university courses was first published in works like "De-Schooling Society" by Ivan Illich that introduced the concept of the use of computer networks as switchboards for learning, in 1970. In 1971 George Kasey, a media(activist)ethicist, delivered a series of lectures on "the Philosophy of Communications De-Design" under the sponsorship of Phil Jacklin PhD, professor at University of California San Jose, a member of "The (San Francisco)Bay Area Committee for Open Media and Public Access." The lectures contained the theoretical outlines for use of telecommunications and media for de-schooling and de-design of mainstream education and an alternative Virtual Free University system. By 1972 George Kasey established "Media Free Times - periodical Multimedia Random Sampling of Anarchic Communications Art" a prototype for remote learning with the use of "multi-media periodicals," that are now commonly referred to as "web pages". In 1995 by John Tiffin and Lalita Rajasingham in their book "In Search Of the Virtual Class: Education in an Information Society" (London and New York, Routledge). It was based on a joint research project at Victoria University of Wellington that ran from 1986-1996. Called the virtual class laboratory it used dedicated telecommunication systems to make it possible for students to attend class virtually or physically and was at first supported by a number of telecommunication organisations. Its purpose was to seek the critical factors in using ICT for university level education. In 1992 the virtual class lab moved onto the Internet and in 1995 The Open University of Catalonia became the first virtual university totally dependent on telecommunications and computers.

A number of other universities were involved in the late eighties in pioneering initiatives and experiments were conducted between Victoria University in New Zealand, the University of Hawaii, Ohio State University and Waseda University to try and conduct classes and courses at an international level via telecommunications. This led to the concept of a Global Virtual University.[3]

2.2 Virtual University System in Rest of the World

Many traditional brick-and-mortar universities have established virtual branches or are at least providing virtual courses. The following is a non-exhaustive list of exclusively virtual universities.

- Canadian Virtual University

  Canadian Virtual University (CVU) is a partnership of Canadian universities collaborating in the development and marketing of distance and online education. CVU is governed by a board of directors, consisting of presidents and directors of distance education at participating universities. President and Chair of the Board is Dr. Lori Wallace, dean of Extended Education at the University of Manitoba. CVU is managed by an executive director, Vicky Busch.

  Following is the link for Canadian Virtual University.

  http://www.cvu-uvc.ca/

- Intercultural Open University

The Intercultural Open University Foundation (IOUF) is an international non-profit charitable foundation that provides a mentored educational experience for post-graduate and graduate learners involved in social change programs around the world.

Following is the link for Canadian Virtual University**.**

http://www.ioufoundation.org/

- Michigan Virtual University

According to its website, Michigan Virtual University® (MVU®) was established in 1998 by the State of Michigan. MVU is a private, nonprofit 501(c)(3) corporation and is governed by an independent Board of Directors composed of individuals representing business, industry, higher education, K-12 education and state government. MVU provides online educational tools, resources and courses for middle and high school students, parents and K-12 educators

Following is the link for Canadian Virtual University.

http://www.mivu.org/

- Rasmussen College

Rasmussen College is a 112-year old for-profit private college offering associate's and bachelor's degrees at multiple campuses in Minnesota, Illinois, North Dakota, Florida, and Wisconsin. It also has an Online Division. In total there are 22 campuses in the Midwest and Florida.

Students may earn diplomas and certificates in a variety of career-focused areas.[5] Rasmussen College also offers on campus and online classes leading to Bachelor of Science (BS), Associate of Applied Science (AAS), and Associate of Science (AS) degrees.

Following is the link for Canadian Virtual University.

http://www.rasmussen.edu/

- Shiraz University

Shiraz University is the first virtual university in Iran. As its name indicates, it was founded in the city of Shiraz. This university serves the people of Iran with online education. It has a big and successful list of Alumni. In addition this university provide traditional brick and mortar system of education.

Following is the link for Shiraz University.

http://shirazu.ac.ir/en/index.php/

- Syrian Virtual University

  The Syrian Virtual University is a Syrian educational institution established by the Syrian Ministry of Higher Education. It provides virtual education to students from around the world. It was established on 2 September 2002 and is the first virtual education institution in the region, and as of 2006, remains the only one. The goals of the SVU include offering education to those who want to learn but cannot afford to do so by going to a "brick and mortar" university. It is headquartered at the Ministry of Higher Education building, Damascus.

  Following is the link for Syrian Virtual University.

  http://www.svuonline.org/

- Virtual University of Pakistan

  The Virtual University of Pakistan (French: Université Virtuelle du Pakistan) is a university based in Lahore, Pakistan. It was established by the Government of Pakistan as a public, not-for-profit institution, with a mission to provide education to students all over the country and abroad. Using free-to-air satellite television broadcasts and the Internet, the virtual university allows students to follow its programs regardless of physical location.

  Following is the link for Virtual University of Pakistan.

  http://www.vu.edu.pk/

- Virtual Global University (VGU)

  The Virtual Global University (VGU) is a private organization founded in 2001 by 17 professors of Business Informatics from 14 different universities in Germany, Austria and Switzerland. The VGU brings together the knowledge and experience of people from different universities in one virtual organization. At the same time it is a real organization, according to German civil law under the name "VGU Private Virtual Global University GmbH".

  Within the Virtual Global University, the School of Business Informatics (SBI) is the organizational unit that offers online courses and an online study program.

  Following is the link for Virtual Global University.

  http://www.vg-u.de/

Chapter 3 Technical Review

3.1 Requirements Gathering

To build a successful system, we should have all the details on requirements of that system. By requirements here we mean who will interact with the system, who will maintain, in which environment it will run, what are the possible problems of users, what is the distribution of authority of use etc.

We used Scrum software development approach (a type of Agile Software development) to gather requirements. We conducted meetings on regular schedule (on daily basis) to discuss what we have previously done, what are the problems and what we need to do next. In software engineering terms such meeting is called *Sprint.* We collected user stories with traditional pen and paper. In each sprint we analyzed those stories and concluded what functionality to add to the system and what to remove. In this way, within a week we had a clear picture of our system.

Now to be more specific in the first sprint decision was made to use IUT as model system and taking all the basic ideas from here and build a virtual system representing this very model. To make more clearer this needs a little elaboration. First of all IUT is an international university for OIC member countries. That means candidates from all member countries of OIC can apply for admissions and candidates other than member states are not eligible. To get admission in IUT every candidate needs to have a twelve year of education with science as major. Our system will reflect these policies. In technical terms we call it business logic. Secondly, IUT has different departments and different programs within each department with well-structured courses for each program. Besides that there is semester system and to promote from one semester to another semester a candidate must qualify a minimum criterion. So there is a preexistent pattern of examinations and all other evaluation. Our system is supposed to take care of all this and other tiny details.

All the requirements will become clearer in the next UML diagrams.

3.2 Use Cases and Diagrams

A use case is a user story explaining how an actor interacts with the system. We use the term actor because the agent who initiates an action can be a student, teacher, administrator or another system. This gives us an overview of the system from the user perspective. The following are the possible use cases:

- Registration for a course
- Select program
- View courses
- Check the schedule
- Go to discussion forum (post questions or comment)
- View academic calendar
- View online students or teachers
- Chat with student or teacher
- View evaluation result
- Download lectures and books
- View gallery
- Login
- Maintenance

Similarly, the system support for interaction of three kind of roles namely,

- Student (Public Users)
- Teacher
- Administrator

In more technical terms it has been shown in UML use case diagram in figure-3.1.

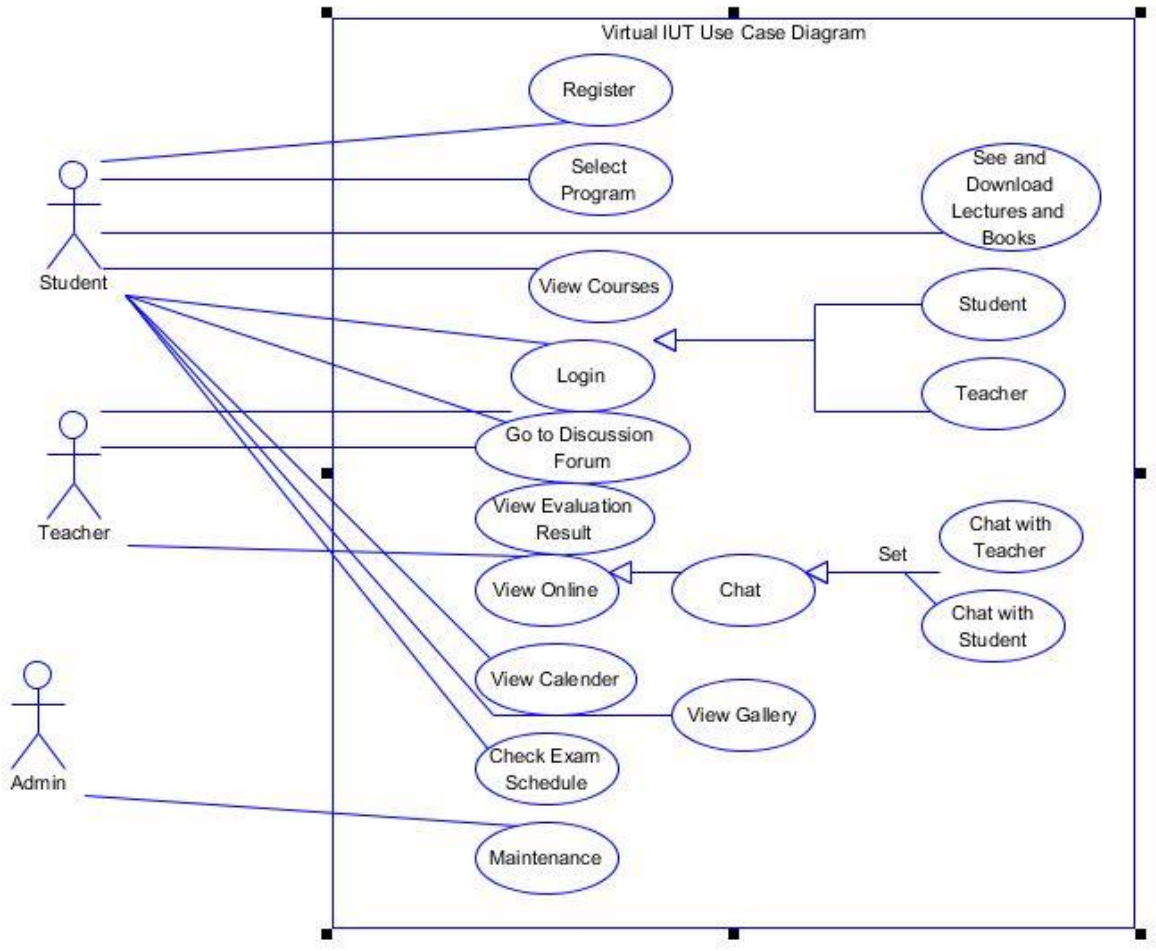

Figure 3.1

3.3 Classes and UML Class Diagrams

To handle the complexity of different operations and take care of business logic we need java beans and model classes in our project. Java beans are supposed to hold data when it is retrieved from the database. Our project follows the famous MVC (Model View Controller) pattern with separate layers for presentation, business logic and persistence data. How the data should appear to user is handled by presentation layer while what constraint should apply to data before it is displayed to user or permanently stored is controlled by business logic and similarly, how data should be permanently preserved is the job of persistence layer.

Following is the list of classes:

- Classes responsible for communicating with the database are listed below. Each one as its name indicates aggregates common functionalities.
  - InsertJava
  - DeleteJava
  - UpdateJava

    - GetJava
- Following are the java beans to keep the data intact once it has been retrieved from the database.
    - DepartmentBean
    - CourseBean
    - StudentBean
    - AssignmentBean
    - Menus
    - BookBean
    - Dept_Course_Bean
    - ProgramBean
    - TopicBean
    - TeacherBean
    - LabBean
    - SemesterBean
    - TeacherCourseBean
    - CommentBean
    - LoginBean
    - DataConnection
- Servlet class which occupies the controller portion of MVC is mentioned below.
    - LoginServlet

Now it's time to show all these classes in UML class diagram to conclude its relationship. Figure. 3.2 is showing the UML class diagram.

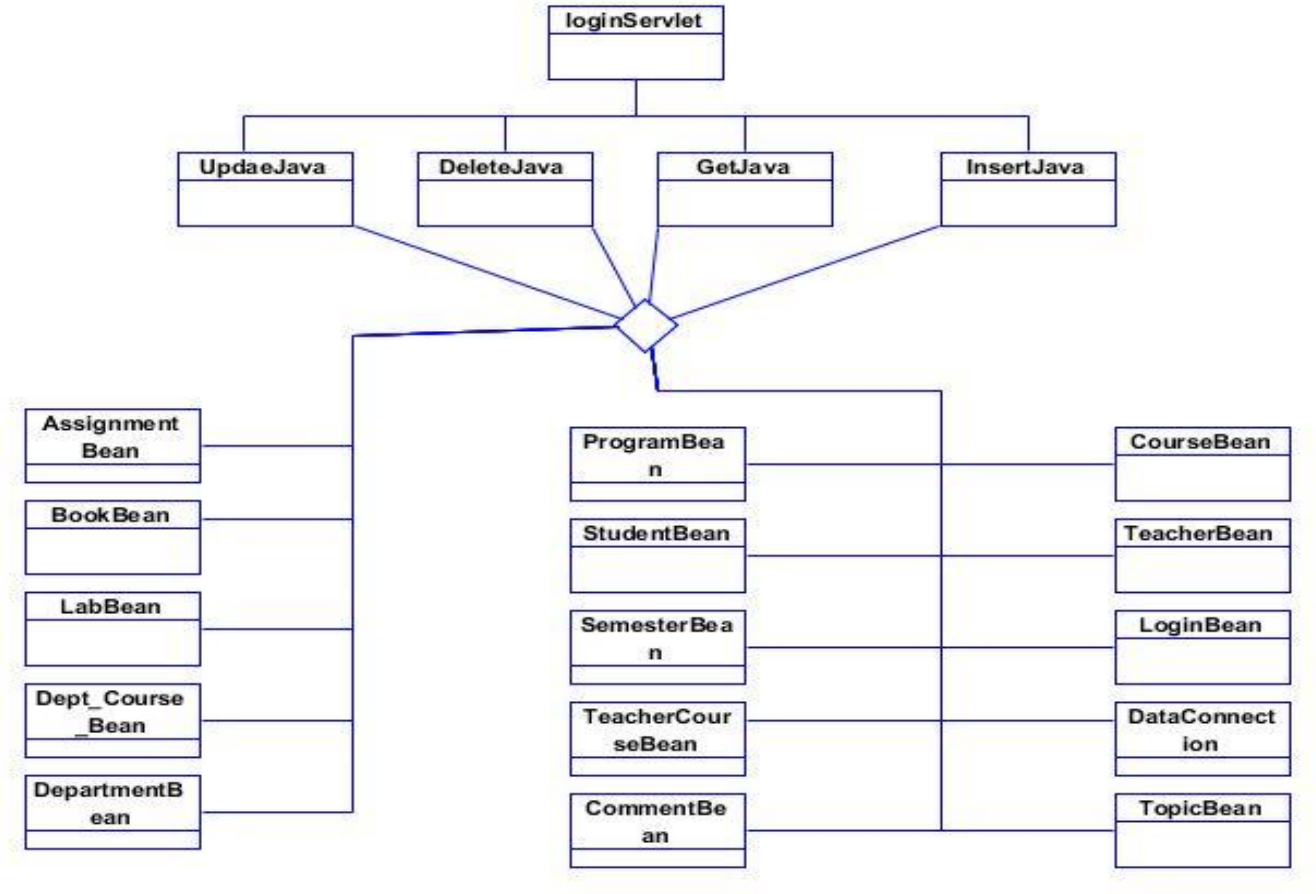

Figure 3.2

3.4 Package Diagrams

Package diagrams combine related other diagrams like use case diagrams or class diagrams. Package diagrams are necessary in case there are too many classes or use cases. It gives us an overview of the system or parts of the system. By seeing this we can have a quick view of how different elements are related to each other.

Figure-3.3 shows the UML package diagram in question.

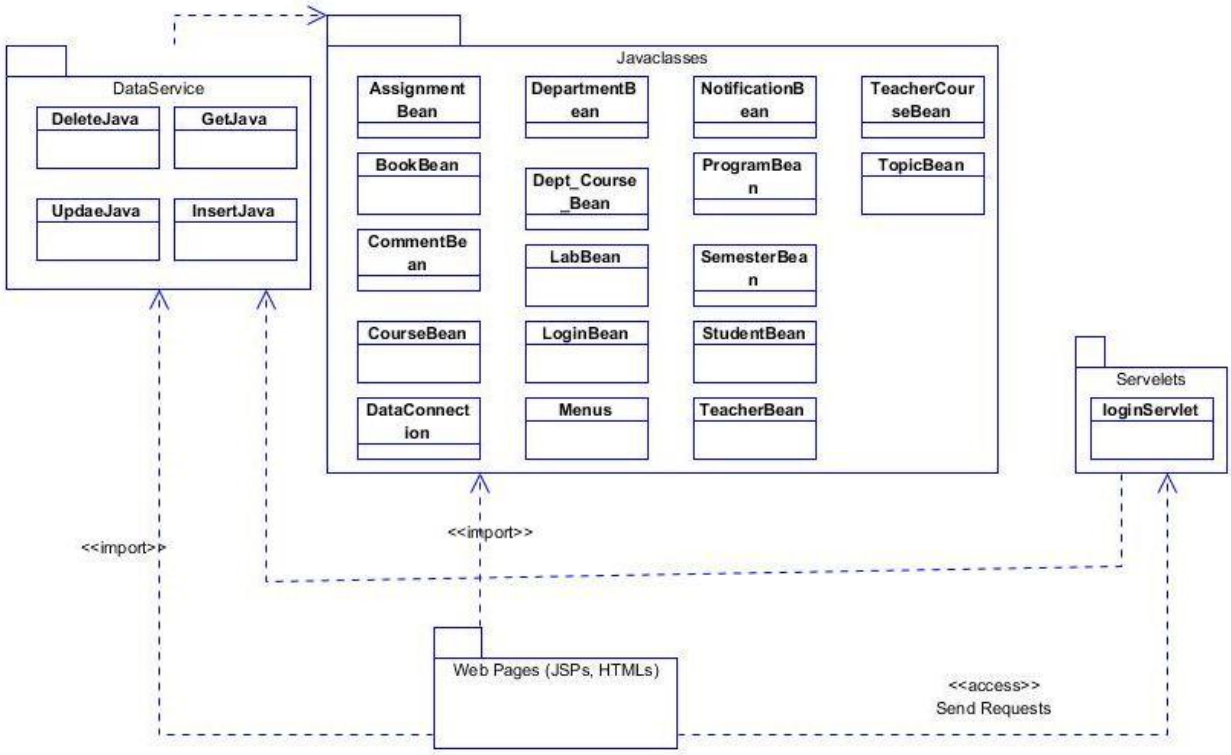

Figure 3.3

3.5 Activity Diagram

Before we go into the details of UML activity diagrams let's take a brief overview of the activity diagrams. Activity diagram shows what happens during a process or operation. They are similar to flowcharts with standard notations to model the flow of data or control during an operation. We use them in modeling phase of generic software development and in elaboration phase of unified process modeling to give a visible shape to business logic. Activity diagram in figure-3.4 showing over all actions of the system.

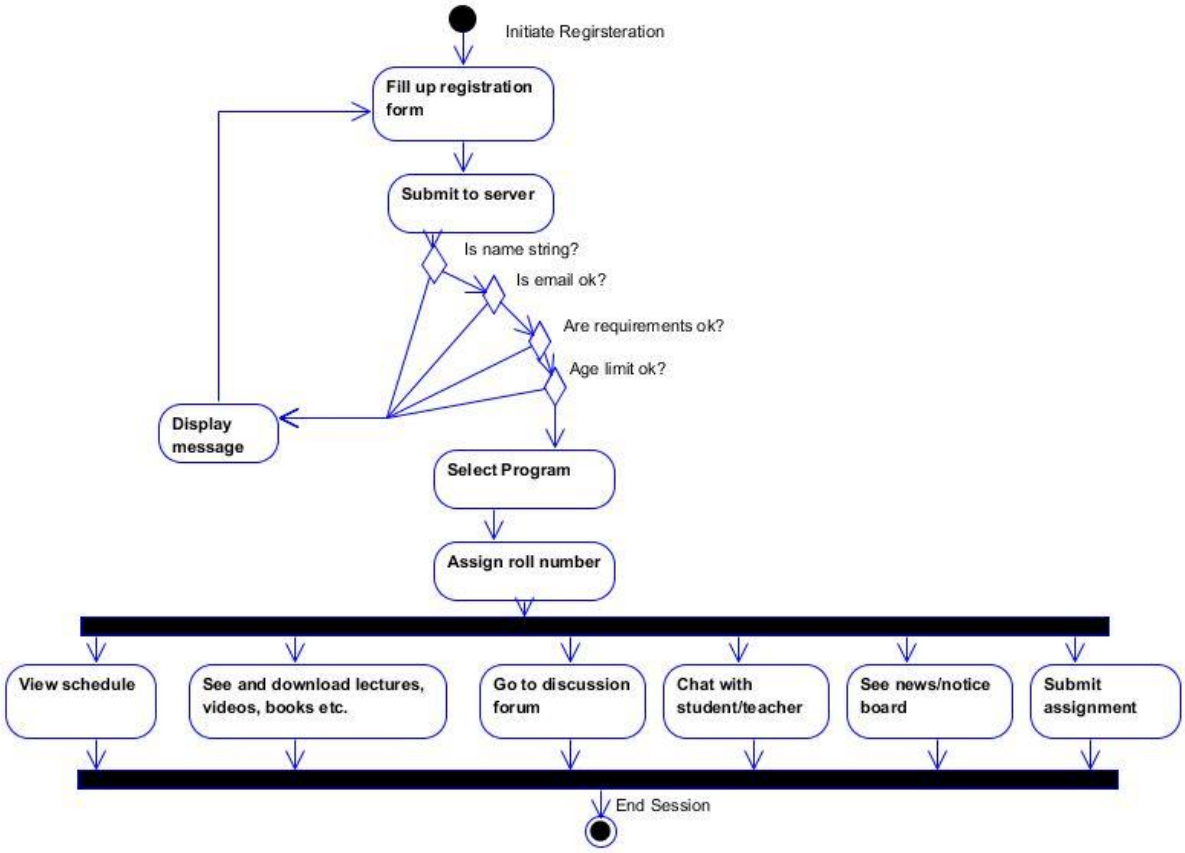

Figure 3.4

3.6 Deployment and Deployment UML Diagrams

Deployment refers to installment of the system in environment where actors directly interact with the system. In this phase software artifacts are mapped onto the hardware. Execution environment is specified and the connection among different nodes is made to enable easy exchange of data. All the components to work require a physical location.

Deployment UML diagrams show hardware to software mapping and hardware and also shows static view of software component and hardware component in their runtime configuration. Besides, it models system's logical components, their physical location and the way they communicate. This discussion will become clear if we see the deployment diagram in figure-3.5.

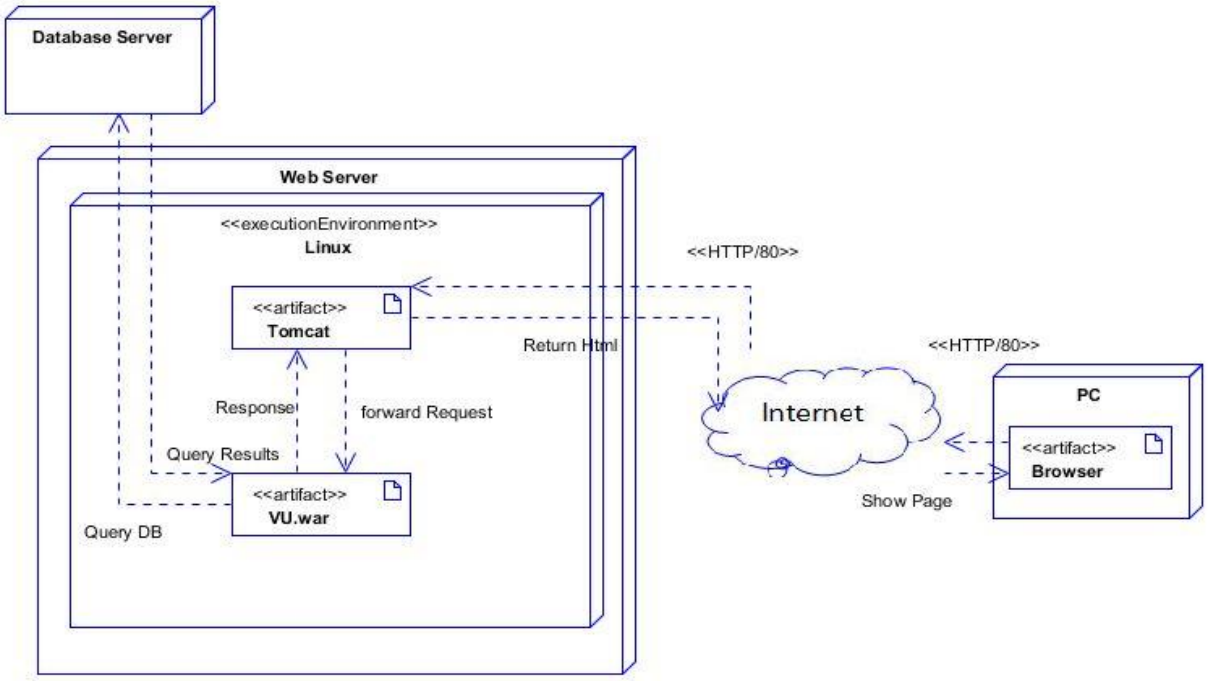
Database Server
Web Server
<<executionEnvironment>>
Linux
<<artifact>>
Tomcat
Return Html
<<HTTP/80>>
<<HTTP/80>>
Response
forward Request
Query Results
<<artifact>>
VU.war
Query DB
Internet
PC
<<artifact>>
Browser
Show Page

Figure 3.5

Chapter 4 Tools and Technologies

4.1 Model-View-Controller (MVC)

Model-View-Controller (MVC) architecture is the architecture that separates content generation and content presentation. Successful use of the pattern isolates **business logic** from **user interface** considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other. MVC divides the whole architecture into three components, View, Model and Controller.

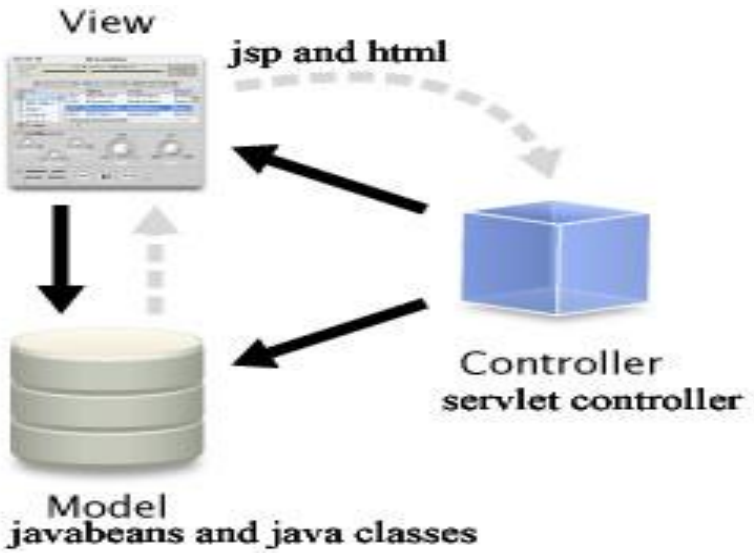

Figure 4.1

1. A controller can send commands to its associated view to change the view's presentation of the model. It can send commands to the model to update the model's state. For example we used Servlet controller.
2. A model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them. For example we used javabeans and java classes as our models.

3. A view requests from the model the information that it needs to generate an output representation. For example we used jsp and html files as our view components.

4.2 Servlet controller

A servlet is a Java class that can be loaded dynamically into and run by a special web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.

When the user requests any file or information from the server, the requests are directed to the *controller* servlet. The controller servlet access required data and builds the *model* and then passes control to the appropriate JSP responsible for presenting the *view*. The user interacts with the controller servlet via the view to access information in the server. Servlet has two main function for handling requests to the server doGet() and doPost() methods.

4.3 Java server page (JSP)

Java Server Page (JSP) is a server-side programming technology that enables the creation of dynamic web pages and applications. This is accomplished by embedding Java code into HTML, XML, DHTML, or other document types. The JSP technology allows the programmers to embed Java code into html (.jsp) page. Java Server Pages are first compiled into Java Servlets by a JSP compiler and then this Servlet is loaded by the Servlet container to server the client request.

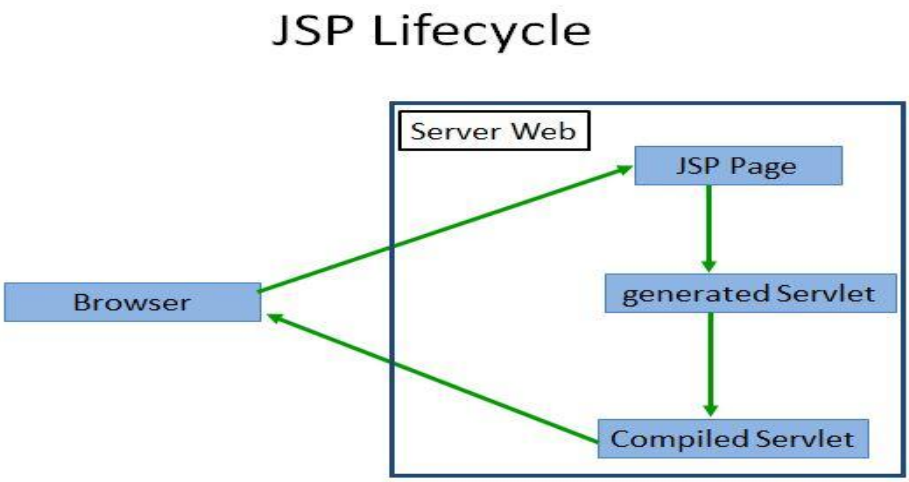

Figure 4.2

Inside the JSP container is a special servlet called the page compiler. The servlet container is configured to forward to this page compiler all HTTP requests with URLs that match the .jsp file extension. This page compiler turns a servlet container into a JSP container. When a .jsp page is first called, the page compiler parses and compiles the .jsp page into a servlet class. If the compilation is successful, the jsp servlet class is loaded into memory. On subsequent calls, the servlet class for that .jsp page is already in memory; however, it could have been updated. Therefore, the page compiler servlet will always compare the

timestamp of the jsp servlet with the jsp page. If the .jsp page is more current, recompilation is necessary. With this process, once deployed, JSP pages only go through the time-consuming compilation process once.

Jsp by default has Implicit Objects predefined like for example Request, response, pageContext, session, application, out, config and page.

Syntactic elements:

<%@ directives %>

<%! declarations %>

<% scriptlets %>

<%= expressions %>

<jsp:actions/>

<%-- Comment --%>

4.4 Javabean (model)

JavaBeans technology is the component architecture for the Java 2 Platform, Standard Edition (J2SE). Practically, they are classes written in the Java programming language conforming to a particular convention. They are used to encapsulate many objects into a single object (the bean), so that they can be passed around as a single bean object instead of as multiple individual objects. Objects. Javabean implements serialisable interface and provide only the setter and getter methods.

We used javabeans for mainly as objects that holds data from the database and we insert data into the object from the database using the setter methods and retrieve the data from the object using the getter methods.

4.5 Custom tags

JSP technology also provides a mechanism for encapsulating other types of dynamic functionality in *custom tags*, which are extensions to the JSP language. Custom tags are usually distributed in the form of a *tag library*, which defines a set of related custom tags and contains the objects that implement the tags.

There are four elements of the JSP specification in particular that address the use of custom tags. They are:

1. A special JSP custom tag API that all tags must obey (e.g., all tags must implement the Tag interface)
2. A strict life cycle and programming model for the tags
3. The Tag Library Descriptor (TLD), which maps tag names to handler classes and provides tag-based syntactic information
4. An extension to the web application deployment descriptor making it possible to point to the tag libraries used within the web application

Custom tag libraries are a powerful feature of JSP that allows Java developers to hide code for database access and other complex operations in custom tags. We have used custom tags for basically displaying data from the database by specifying a particular tag in our jsp files.

4.6 Scripting languages

Scripting languages are class of component programs that are either executed in the client side, by the user web browser or executed by web server in the server-side.

4.6.1 Client-side Scripting

Client-side scripting is a class of component program on the web that is executed in the client-side, by the user's web browser, instead of server-side (web server). In our project we have used html (hypertext markup language), javascript, jquery and Ajax as our scripting languages. When a request is sent to the server the necessary file are sent to the user's computer on the browser on which they reside. These client-side languages are then executed on the user browser.

4.6.1.1 Html and Javascript

The user computer downloads the html/javascript files on the web browser, reads and then executes the html/javascript tags.

We used JavaScript to validate form data before we submitted it to a server(TOMCAT). This saves the server from extra processing. And also we used javascript to react to certain events, like hiding certain tags and displaying them when a particular event occurs.

4.6.1.2 Jquery

jQuery is a library of JavaScript Functions. It has many Ajax and Javascript features that allow you to enhance user experience and semantic coding. We first download a copy of JQuery and then we insert it in our html or jsp page (preferably within the <head> tag). Then we write functions to tell jQuery what to do. Like for example, we used it to generate the popup dialog box, glowing menus, glowing popup gallery pictures and extra.

4.6.1.3 Ajax

AJAX is an acronym for Asynchronous JavaScript and XML. AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. Ajax essentially offers a technique for client-side JavaScript to make background server calls and retrieve additional data as needed, updating certain portions of the page without causing full page reloads.

We used Ajax to basically pass our request like update, insert or retrieval of our information from the server to our client browser without necessarily reloading the whole page. This has also played a big role in simplifying the work of the user and in providing dynamicity to our project.

4.7 Graphics Design and Editing Tools

For designing user interface, graphic tools were used. There are various graphic tools currently and we have tried to our level best to use these tools. Among the tools we used include.

4.7.1 Macromedia Firework

This is a Microsoft product and usually comes with macromedia DreamweaverMacromedia Fireworks 8 is the definitive solution for professional web graphics design and production. We used Fireworks to create, edit, and animate web graphics, add advanced interactivity, and optimize images in a professional environment.

We used it in our project to design banner images, logos, editing gallery images and other images that required editing.

4.7.2 Snipping Tool

Snipping Tool is used to capture a screen shot, or snip, of any object on your screen, and then annotate, save, or share the image. It's a free tool that comes with every operating system.

This tool played a big role in capturing the screen shots of our projects, databases, diagrams and other images.

4.7.3 Edraw Max v4.5

Edraw Max is a professional tool for designing advanced system diagrams like for example designing data flow diagrams, Entity relationship diagrams, decision tree diagrams, data dictionaries, input process and output(IPO) diagrams and visual table of content (VTOC) diagrams.

In our project we used Edraw max to draw and design Entity relationship diagrams and decision tree diagrams.

4.7.4 Visual Paradigm for UML

Visual paradigm for UML is a UML CASE Tool supporting UML 2,SysML and Business Process Modelong Notation (BPMN) from the Object Management Group (OMG). There are many editions used but we used a free edition called Community Edition.

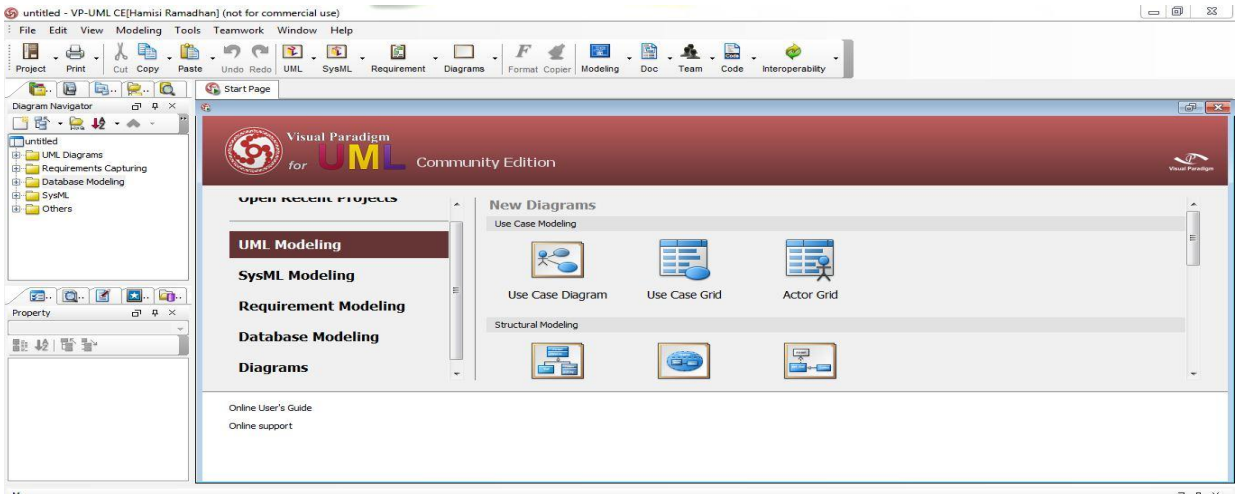

Figure 4.3

VP-UML has many features among them include UML Diagrams, SysML Diagrams, Requirements Management, Business Process Modeling, and Database Modeling.

We used VP-UML to mainly design our UML diagrams which include Class diagram, Use case diagram, Sequence diagram, Activity diagram, Component diagram, Deployment diagram, Package diagram and Object diagram.

4.8 Database Used

In our project we used a server based database known as MySQL. The version of MySQL used was MySQL 5.1.35- community via TCP/IP.

A database server refers to the back-end system of database application using client/server architecture. This back-end performs tasks such as data analysis, storage, data manipulation, archiving, and other non-user specific tasks.

MySQL provides both GUI(Graphical User Interface) and command interfaces. Either way

,it can be integrated with netbean project by including mysql drivers in netbean and to access MySQL in the project we include MySQL connector in the project libraries.

4.9 Web Browsers

During website development, a page might load perfectly in one browser and fail in another. In this regard, we decided to test our project in different browsers to check whether its being supported by them. Among them include;

- Mozilla firefox
- Internet explorer
- Google crome

- Opera
- cometbirds

Chapter 5 Database Design

In MVC pattern the last layer is database where data is stored permanently. This section requires a full attention on its own. A very professional touch is needed to design this part. As far our design is better that far our data becomes meaningful and easy to extract. There are various database designs but the one we selected is relational database and MySql as the relational database management system in which data is stored in relations or tables.

The first step in designing database is to identify independent entities in the system. Then categorize whether it is strong or weak entity. Secondly, we determine the list of the attributes. Among the attributes we determine which attribute uniquely identify an entity to use it as primary key. We also need to know whether an attribute is composite or derived attribute and act accordingly. Now the crucial part of design process is determining relationships among entities. The four possible relationships are one-to-one, one-to-many, many-to-one and many-to-many.

5.1 ER Diagram

To meaningfully represent what we just discussed we use entity relationship diagram (ERD). ERD is a collection of symbols and notations which convey some meaning. Before we show our ERD we will introduce the basic notations which are:

- Entity is represented by a rectangle.
- Attribute by an ellipsis.
- Relationships by a diamond shaped symbol connecting two entities.

Figure-5.1 shows the ERD for our system.

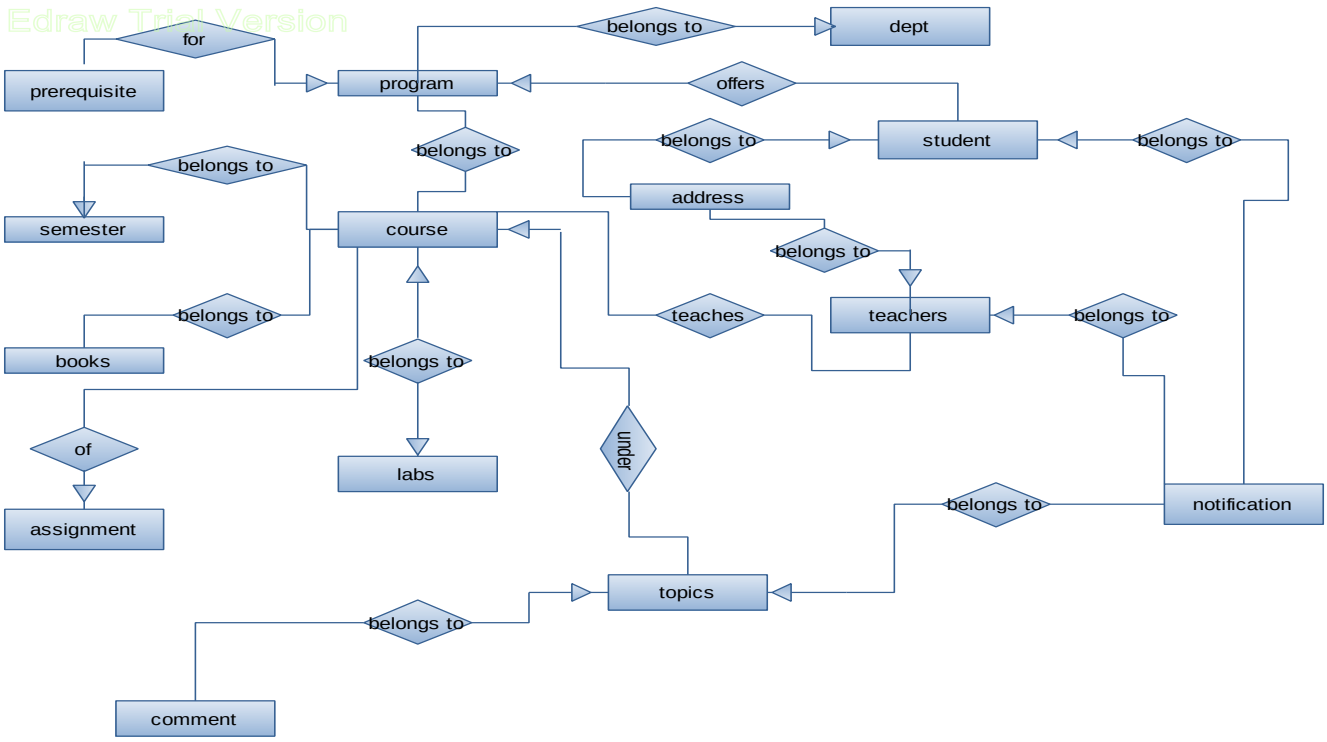

Figure 5.1

5.2 Relational Schema

5.2.1 Schemas derived from a strong entity

Dept= ((**Did**, dname, dhead, description))

Program = (**Pid**, pname, duration, NOS)

Student= (**Stid**, stdname, email, contact, DOB, fname, mname, RedDate, current)

Teacher= (**TID**, Tname, Qualification, contacts, email)

Course = (**CID**, Cname, credit)

Semester= (**SID**, sname)

Topic = (**Td**, Tdate, TopicTitle)

Notificationz = (**nid**, notification)

Books = (**Bid**, bname,author, edition)

Assignment = **(Asid**, title, marks)

Login = (**Username**, password,Level, status)

5.2.2 Schemas derived from multiple attribute

Book_course = (**Bid, CID**)

Dept_course = (**CID, Did,** Cname, credit SID)

Teacher_course = (**TID, CID**)

5.2.3 Schemas derived from relationship set involving strong entity sets

Program_student = (**Pid**, Stid)

Dept_program= (**Did**, Pid)

Topic_comment= (**Tp**, Cd)

Course_topic = (**CID**,Tp)

Course_teacher = (**CID, TID**)

Program_course = (**Pid, CID**)

Course_semester = (**CID, SID**)

Course_books= (**Bid,**CID)

Course_assignment = (**Asid,**CID)

Notificationz_topic = (**Tp ,**nid)

Notificationz_student = (**Stid**, nid)

Notificationz_teacher = (**TID,** nid)

5.2.4 Schemas derived from weak entity set

Lab = (Lid, labname)

Comment = (Cd, cdate, content, user)

5.3 SQL Code

Structured Query Language is the tool to create database, tables and populate with data. As we saw relational schema, based on that we have pretty much good idea how our tables in database would look like. We can't give all DML queries that we used because it will grow our discussion here larger. So here are the queries that used to create tables with relationships.

5.4 Snapshots from MySql

To have a practical look at database we have taken some snapshots of MySql in action and are very eager to make it a part of this document. As by now you know what tables are there in our database. This will also provide an overview of the data in database.

5.4.1 Book Table:

Figure 5.2

5.4.2 Course Table

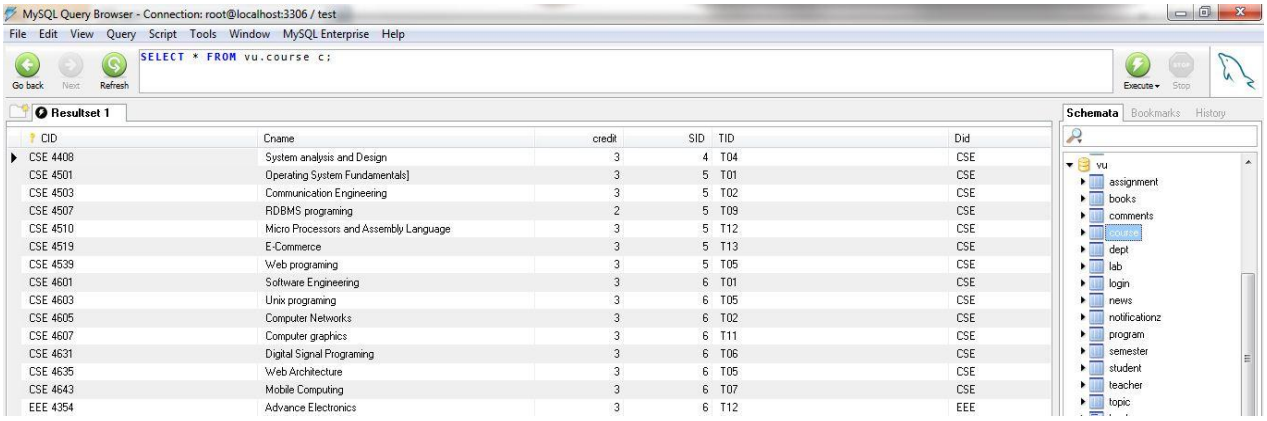

Figure 5.3

5.4.3 Department Table

Figure 5.4

5.4.4 Student Table

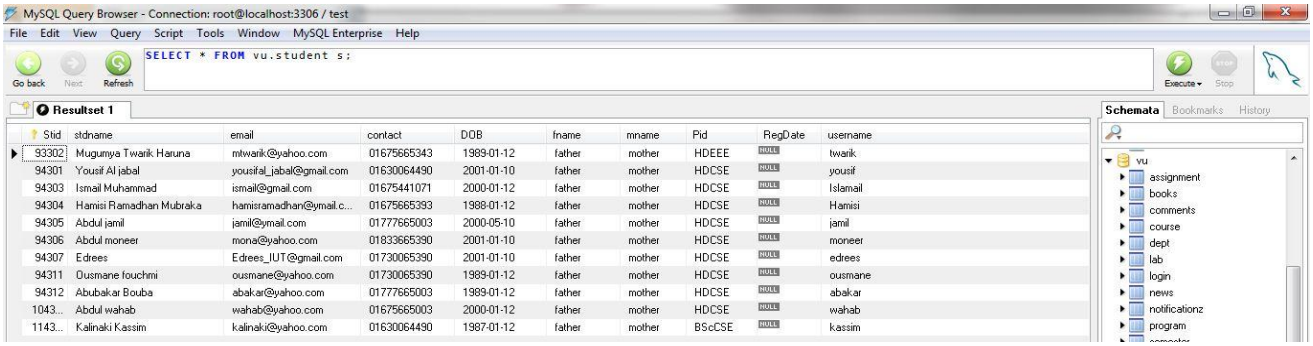

Figure 5.5

5.4.5 Program Table

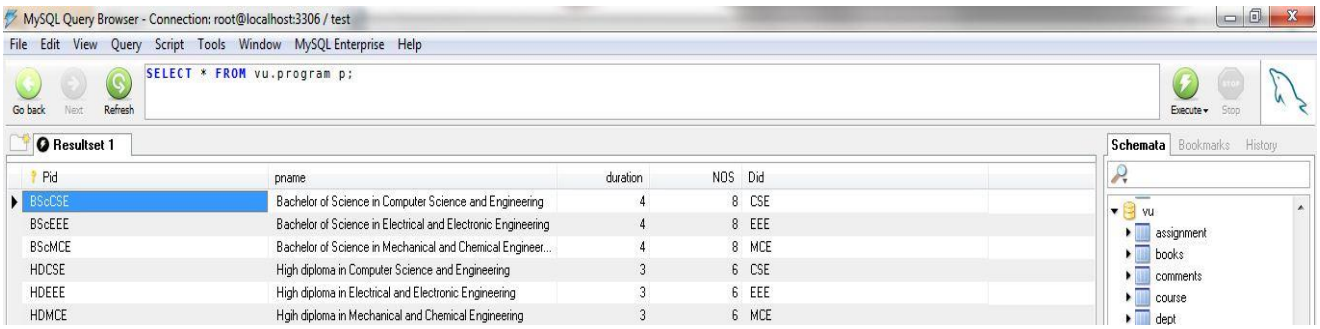

Figure 5.6

Chapter 6 Functionalities

To have a user look to the system, we must shed some light on its functionalities. Functionalities are services available to the user. By reading this chapter one can easily have an idea of the site without visiting it.

6.1 Web 2.0 Standard Features

Web 2.0 is a concept that takes the network as a platform for information sharing, interoperability, user-centered design, and collaboration on the World Wide Web. A Web 2.0 site allows users to interact and collaborate with each other in a social media dialogue as creators (prosumers) of user-generated content in a virtual community, in contrast to websites where users (consumers) are limited to the passive viewing of content that was created for them.

We managed to integrate some of the web 2.0 standards in our project and we shall elaborate them in detail. We have divide web 2.0 standard used in this project into modules; chatting, discussion and email module.

6.1.1 Chatting facility module

In most of the physical universities the teacher and the student communicate orally with each other. In contrast to our virtual university, there is no real time direct communication between students and teachers. In this regard, there is a great need to develop a mean by which a student can ask a teacher or a fellow student directly and receive a response in real time.

In this project we also included a chatting mechanism by which students can interact with their fellow student and with teachers as well.

When a student logs into our university site he is directed to his home page as shown below.

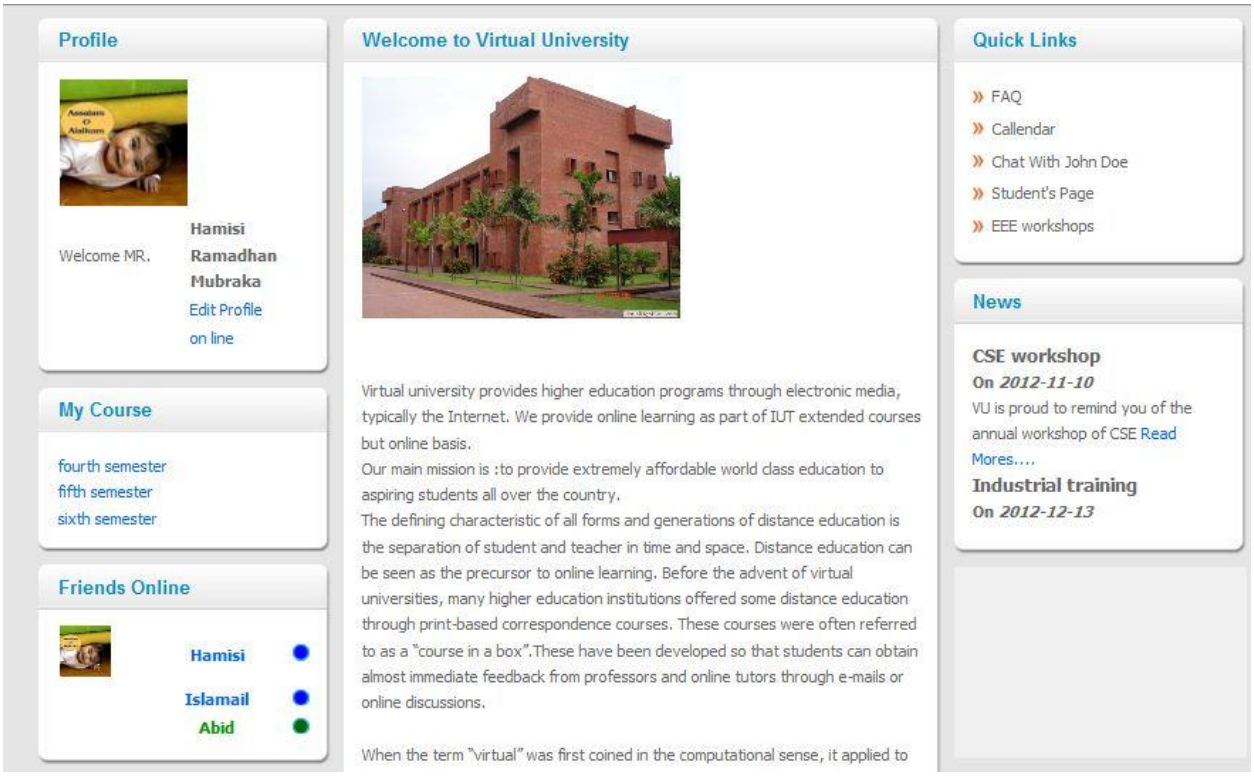

Figure 6.1

When a student logins in, the site recognizes him by displaying his profile detail to his left. And bellow "my course" block is a friend's online block. In this block all his class mates and teachers online will be displayed visible to him. And from here he can chat directly with any of them.

This enhances and improves communication between members and the can privates share information without being intervened.

6.1.2 Discussion module

As a matter of fact in every university or train institute there is a provision where students are grouped together in to a discussion. It's from here where student share ideas and know with each other.

In our project we also included a discussion facility where students can ask a question and it can be discussed upon by all class members. We divided our discussion basing on the courses offered at the university, that is to say, every course has a list of discussion topics to which students can participate in.

When a student selects a particular course he is directed to a discussion page of that course to which he chooses a topic of interest. From here the ideas of different members is displayed, as shown below.

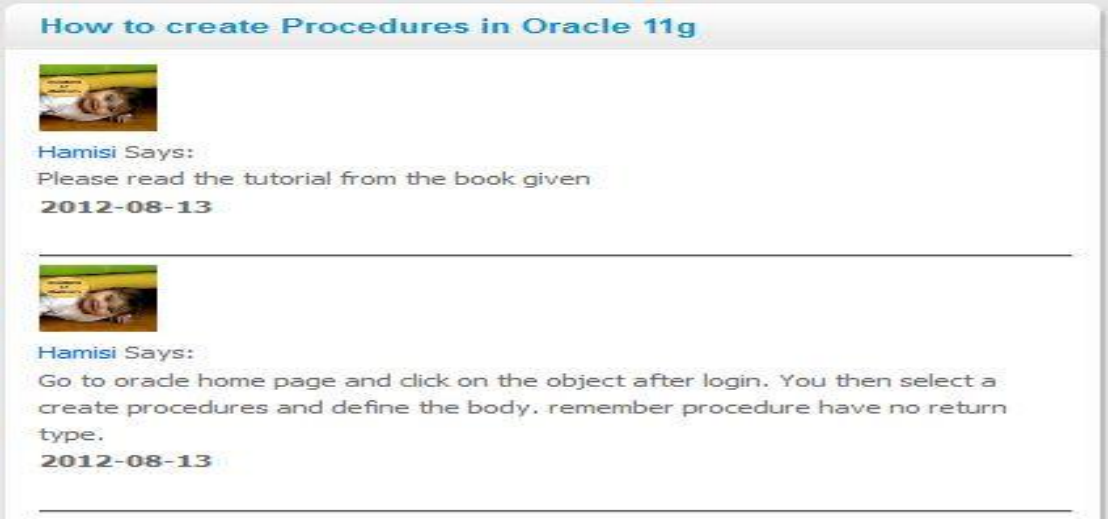

Figure 6.2

Above is a snap short of a portion of discussion page. A student profile picture and his comment is display. This discussion can only be visible to only student of same class and teacher of that course.

6.1.3 Email facility

Though there are dozens of free email service providers out there, one may ask why we need to such a feature here. Well, this can be useful in many cases like there is contact information available on site, someone may want to ask about any problem from authorities. So in that case copying going to a third party and using its service for sending email is a little longer and cumbersome process. She/he can directly send from here without going somewhere else. Besides that email is a relay messaging. That means a student might have some confusion on his mind to ask some teacher which might not be online at the moment, so he can at his/her convenience send him/her an email and wait for reply. Someone may be interested to send a simple email to a person outside the domain of the university, in that case he can also use this facility.

We are using javamail api for sending email. The recipient will see the address from which the mail was sent to him and any reply to mail will come to that address not to a university dedicated mail server. To dig a little more deep, we use SMTP for sending mail but SMTP server at the same time is not a dedicated one. Instead, we are using Gmail SMPT server to accomplish the task. Following is a snap shot from the email interface.

Figure 6.3

6.2 Academic Modules

This module is divided by three parts; Student academic module, Teacher academic module and Administrator academic module.

6.2.1 Student Academic Module

When a student logs in, he can do the following;

6.2.1.1 View Course Details

When a student logins into the system all the courses he has done up to the current semester courses will be displayed. The courses will be displayed in semester wise format. All course of that semester will be displayed beneath the semester and when the user hovers over the semester it will display them. This is as shown below.

Figure 6.4

Above is a snap short of part of a project displaying courses offered by the respective student.

When a student click on a particular course the content details of that course will be displayed besides it. Like for example when he clicks on course CSE 4635 details of the course will look like this.

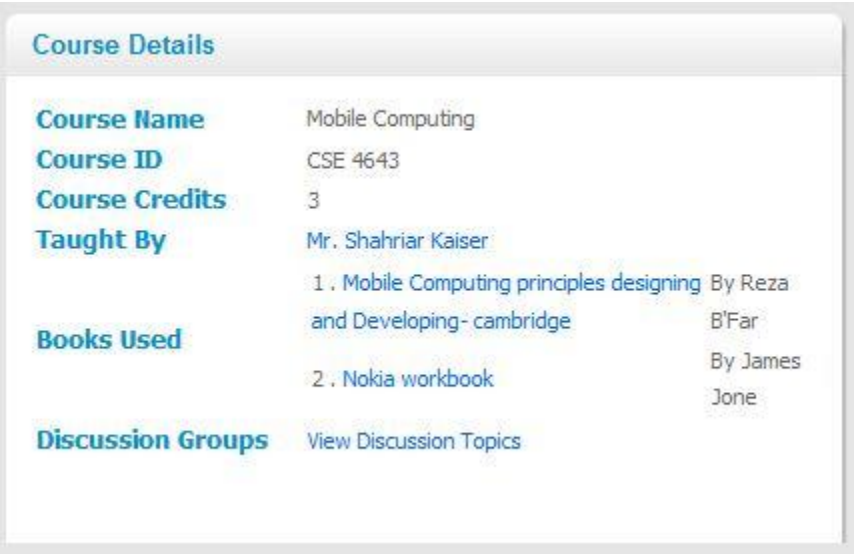

Figure 6.5

As shown above, a student will be able to view all course detail including the discussion groups.

6.2.1.2 Viewing teacher's details

When a student clicks on a particular teacher, the teacher details of that particular teacher will be displayed. This will be shown in a popup format for simplicity to avoid too much reloading. As shown below.

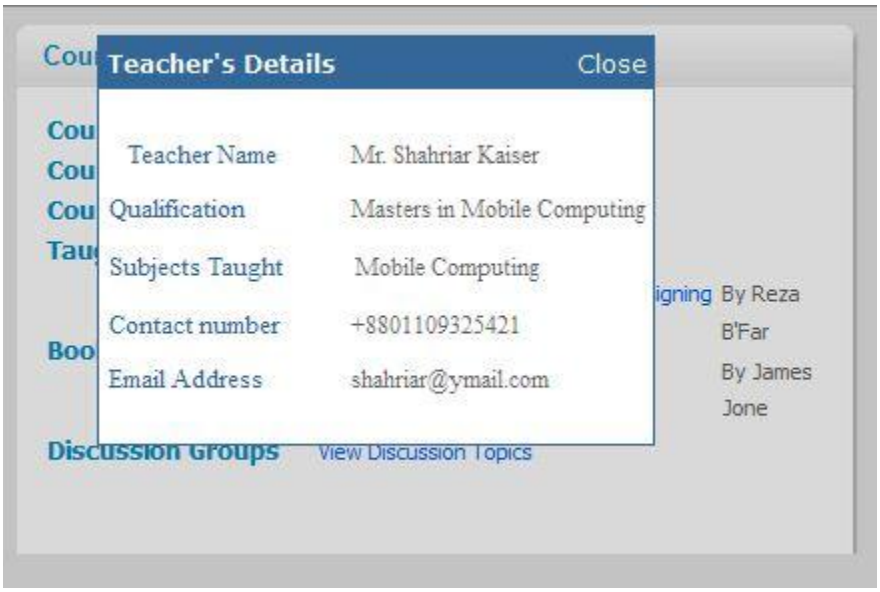

Figure 6.6

As shown above, the teacher's name, qualification, the subject he teachers, his contact number and the mail address will be displayed.

6.2.1.3 Read Books

For every course there are books referred by the teacher. These books a student can obtain from anywhere he/she wishes at his/her ease in hard copy or soft but university is also providing respective books right on the site for reading or downloading.

6.2.1.4 Download Lectures

The lecture schedule is disseminated before just before the start of the semester. As soon as there is a new lecture available, all of its respective details like powerpoint slides, tutorials, references or pdf files are uploaded and are ready to download.

6.2.2 Teacher Academic Module

There is not much on the teacher side except that when teacher have all the materials for a lecture available then he/she can upload in order to be download by all of its students. Among the materials that a teacher can upload include videos, ppt slides, pdf files and references to other tutorials across the web. Besides that a teacher can participate in live chat or discussion forum in order to help teacher in case if they face any sort of problem.

6.2.3 Admin Academic Module

Admin is the person responsible for most of the plug and play in the system. Admin has access to all information of the students as well as teachers. Admin will be the person who will have command over database transaction handling as well as technical details of the web applications. There is no dedicated interface for admin to carry out its maintenance job but we do assume admin as a professional. Admin is the person who can introduce changes to the site like changing interface, deleting materials, adding, doing more complicated things which a teacher can't do.

6.3 Other Utilities

Besides very basic functionalities needed for distance learning, there are some utilities which makes the student feel at home when they visit the site. Among them are:

6.3.1 Academic Calendar

Academic calendar is such a utility which keeps track of all important events like convocation, important examinations etc.

6.3.2 Gallery

Galleries give a pictorial view of the different aspects of the universities, alumni, important events and functions. Figure-6.7shows a snapshot of a gallery. In the figure links to the different galleries are show. As picture is clicked, it activates jquery which pops up the picture and display it to the user in most attractive way.

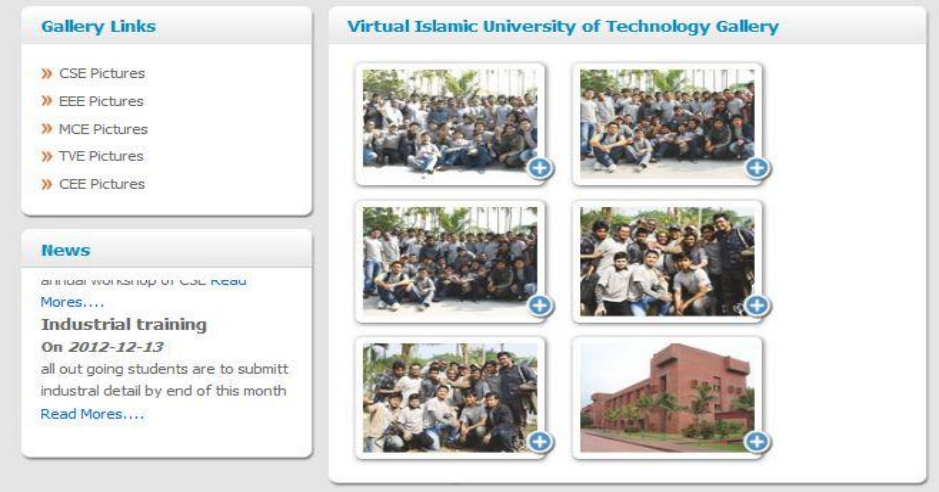

Figure 6.7

6.3.3 News Board

There is a portion dedicated to news. It shows title the news and a link to its details. News are moving up within the board and by hovering mouse over it, it stops. Figure-6.8 shows how news board looks like. Teacher can also put different notifications on the news board.

Figure 6.8

Once someone click on the "*Read More. .*" link it will display the detail news board. In figure-6.9 we show a view of the detailed news board.

Figure 6.9

6.3.4 FAQs

There are always some common problems whose solution is needed for everyone. FAQ portion combines all those questions which can appear in minds of all new students.

Chapter 7 Future Planning

It's an old saying that "*Rome was not built in a day*". The same way we had a lot on our mind but to do everything great requires time and resources. Therefore, we do have future targets. By analyzing future technology trends and most commonly available technologies, we determined our targets. In this chapter we discuss them one by one.

7.1 Live Streaming Video Lectures

Live classes are virtual counterparts of the traditional brick-and-mortar classes. The idea is simple. Record and feed to website. A teacher in some remote place is taking class and that is fed live across the Internet where student logged in from different remote places can attend the lectures. This is definitely a perfect idea and our plan is implement it.

7.2 Computer-aided Assessment

Less commonly referred to as E-assessment, ranging from automated multiple-choice tests to more sophisticated systems is becoming increasingly common. With some systems, feedback can be geared towards a student's specific mistakes or the computer can navigate the student through a series of questions adapting to what the student appears to have learned or not learned.

The best examples follow a Formative Assessment structure and are called "Online Formative Assessment". This involves making an initial formative assessment by sifting out the incorrect answers. The author/teacher will then explain what the pupil should have done with each question. It will then give the pupil at least one practice at each slight variation of sifted out questions. This is the formative learning stage. The next stage is to make a Summative Assessment by a new set of questions only covering the topics previously taught.

This is clear from above discussion that there definitely are techniques which we can employ and get our desired purpose. We are planning dig deep into these techniques and exploit them for our system in order to minimize the dependency on traditional system of examination and increase the amount of virtuality.

7.3 Learning Management System and Learning Content Management System

A learning management system (LMS) is software used for delivering, tracking and managing education. LMSs are of set software for managing educational materials, distributing courses over the Internet and offering features for online collaboration. A LMS allows for teachers and administrators to track

attendance, time on task, and student progress. Parents can log on to the LMS to track grades. Students log on to the LMS to submit homework and to access the course syllabus and lessons.

A learning content management system (LCMS) is software for author content (courses, reusable content objects). An LCMS may be solely dedicated to producing and publishing content that is hosted on an LMS, or it can host the content itself.

Though both LMS and LCMS are already in use in parts of the world but our planning is to keep everything organize and traceable are planning to utilize this technology in futures.

7.4 Online Lab

The online lab model delivers an entire course to a student digitally, but within the confines of a brick-and-mortar lab. The students are supervised, but not actively "taught" within the classroom. There are many subjects that do need complementary labs. Therefore, we are also seeking tools to eradicate this deficiency.

7.5 Use of Free Web Based Tools

There are many tools these days available online free of cost which could enhance the process of learning and reduce the cost. Our future planning also involves availing these tools as much as possible. Following is a list of some of them.

- *Social Media:*

  Nowadays, social network are the most popular web services on the Internet and has got the most number of users. Best examples are twitter and facebook.

- *YouTube:*

  Youtube is a free video sharing service. Youtube provides channels for different user defined categories which keeps together videos related to one subject.

- *Google Reader:*

  It is a free service for reading a wide variety of files or documents without downloading them. Most popular is pdf file format.

- *Google Docs:*

  Free document hosting service provided by Google Inc.

- *Slideshare*

  SlideShare is a Web 2.0 based slide hosting service. Users can upload files privately or publicly in the following file formats:PowerPoint, PDF, Keynote or OpenOffice presentations. Slide decks can then be viewed on the site itself, on hand held devices or embedded on other sites. This could be helpful for sharing lectures.

7.6 Video Chat and Conference

With increase of Internet speed and advancement in technology, video chat or video conference has become buzz words in the Internet community and especially in Business Corporations. In scenario of virtual education this can be helpful in many ways like interviewing students, meeting of remote teachers or live classes.

7.7 Efficient Search Engine

People's habits of browsing are constantly changing. As the size of Internet grows larger and larger people becomes more and more dependent on search engines which make them rely heavily on searching facilities. As for now there is no searching capability in our system but in very recent future our plan is a robust search engine.

7.8 RSS Feed

According to many Internet surveys people usually don't like rss feeds but as we are talking in academic perspective so rss feeds could be very helpful quick and important news.

Chapter 8 Conclusion

The purpose of technology is make human life easier and productive. We have technology but requires engineering to use it solve various human problems. Our little effort is the reflection of this thinking. In a short span of time we did something that can cause revolution in education industry. As we mentioned in our future planning that there are many technologies which we can use for our system to serve the purpose in the best way possible. A lot more is required to do to raise it to global standard.

References

[1]. Anderson, M. ' Virtual universities — future implication for students and academics'. *IBM Global Services Australia.*
http://www.ascilite.org.au/conferences/brisbane99/papers/anderson.pdf

[2]. Ahmad, R., Piccoli, G. (1998) 'Virtual learning environments: an information technology basic skills course on the web'. *Association for Information Systems 1998 Americas Conference.*
http://www.isworld.org/ais.ac.98/proceedings/home.html

[3]. Freimut Bodendorf, Philip H. Swain Virtual Universities in Engineering Education
http://www.ijee.ie/articles/Vol17-2/IJEE1241.pdf

[4]. Harasim, L., Hiltz, S., Teles, L., Turoff, M. (1995) Learning Networks: *A Field Guide to Teaching and Learning Online, United States of America: Massachusetts Institute of Technology*

[5]. Java Servlets and JSP by Joel Murach and Andrea Steelman 2nd Edition.

[6]. Web Application Architecture by Leon Shklar and Richar Rosen.

[7]. Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions by Budi Kurniawan.

[8]. Java Servlet Programming, 2nd Edition by Jason Hunter and Willian C.

Appendix 1

Here is the SQL code for the database that we created for the VIUT.

Tables

```
--

-- Create schema vu

--

CREATE DATABASE IF NOT EXISTS vu;

USE vu;

--

-- Temporary table structure for view `book_course`

--

DROP TABLE IF EXISTS `book_course`;

DROP VIEW IF EXISTS `book_course`;

CREATE TABLE `book_course` (

  `Bid` int(10) unsigned,

  `CID` varchar(10)

);

--

-- Temporary table structure for view `dept_course`

--

DROP TABLE IF EXISTS `dept_course`;

DROP VIEW IF EXISTS `dept_course`;

CREATE TABLE `dept_course` (

  `CID` varchar(10),

  `Cname` varchar(45),

  `credit` int(10) unsigned,

  `SID` int(10) unsigned,

  `Did` varchar(10)

);
```

```
--
-- Temporary table structure for view `teacher_course`
--
DROP TABLE IF EXISTS `teacher_course`;
DROP VIEW IF EXISTS `teacher_course`;
CREATE TABLE `teacher_course` (
  `TID` varchar(10),
  `CID` varchar(10)
);
--
-- Definition of table `assignment`
--
DROP TABLE IF EXISTS `assignment`;
CREATE TABLE `assignment` (
  `Asid` varchar(10) NOT NULL,
  `title` varchar(45) NOT NULL,
  `marks` int(10) unsigned NOT NULL,
  PRIMARY KEY (`Asid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Definition of table `books`
--
DROP TABLE IF EXISTS `books`;
CREATE TABLE `books` (
  `Bid` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `bname` varchar(245) NOT NULL,
  `author` varchar(45) NOT NULL,
  `edition` int(10) unsigned NOT NULL,
  `CID` varchar(10) NOT NULL,
  PRIMARY KEY (`Bid`),
```

```
  KEY `FK_books_Course` (`CID`),

  CONSTRAINT `FK_books_Course` FOREIGN KEY (`CID`)
REFERENCES `course` (`CID`) ON DELETE CASCADE ON UPDATE
CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

--

-- Definition of table `comments`

--

DROP TABLE IF EXISTS `comments`;

CREATE TABLE `comments` (

  `Cd` int(10) unsigned NOT NULL AUTO_INCREMENT,

  `cdate` datetime NOT NULL,

  `Td` int(10) unsigned NOT NULL,

  `user` varchar(45) NOT NULL,

  `content` text NOT NULL,

  PRIMARY KEY (`Cd`),

  KEY `FK_Comments_topic` (`Td`),

  CONSTRAINT `FK_Comments_topic` FOREIGN KEY (`Td`)
REFERENCES `topic` (`Td`) ON DELETE CASCADE ON UPDATE
CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;

--

-- Definition of table `course`

--

DROP TABLE IF EXISTS `course`;

CREATE TABLE `course` (

  `CID` varchar(10) NOT NULL,

  `Cname` varchar(45) NOT NULL,

  `credit` int(10) unsigned NOT NULL,

  `SID` int(10) unsigned NOT NULL,

  `TID` varchar(10) DEFAULT NULL,

 `Did` varchar(10) DEFAULT NULL,
```

```
  PRIMARY KEY (`CID`),

 KEY `FK_Course_Semester` (`SID`),

  KEY `FK_course_Teacher` (`TID`),

  KEY `FK_course_dept` (`Did`),

  CONSTRAINT `FK_course_dept` FOREIGN KEY (`Did`)
REFERENCES `dept` (`Did`),

  CONSTRAINT `FK_Course_Semester` FOREIGN KEY (`SID`)
REFERENCES `semester` (`SID`) ON DELETE CASCADE ON UPDATE
NO ACTION,

  CONSTRAINT `FK_course_Teacher` FOREIGN KEY (`TID`)
REFERENCES `teacher` (`TID`) ON DELETE CASCADE ON UPDATE
CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Definition of table `dept`

--

DROP TABLE IF EXISTS `dept`;

CREATE TABLE `dept` (

  `Did` varchar(10) NOT NULL,

  `dname` varchar(45) DEFAULT NULL,

  `dhead` varchar(45) DEFAULT NULL,

  `description` varchar(245) DEFAULT NULL,

  PRIMARY KEY (`Did`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Definition of table `lab`

--

DROP TABLE IF EXISTS `lab`;

CREATE TABLE `lab` (

  `Lid` int(10) unsigned NOT NULL AUTO_INCREMENT,

  `labname` varchar(45) NOT NULL,

  PRIMARY KEY (`Lid`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Definition of table `login`
--
DROP TABLE IF EXISTS `login`;
CREATE TABLE `login` (
  `Username` varchar(20) NOT NULL DEFAULT '',
  `Password` varchar(45) NOT NULL,
  `Level` varchar(45) NOT NULL,
  `status` int(10) unsigned DEFAULT NULL,
  PRIMARY KEY (`Username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Definition of table `news`
--
DROP TABLE IF EXISTS `news`;
CREATE TABLE `news` (
  `idNews` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NewsDate` date DEFAULT NULL,
  `NewsHeading` varchar(45) DEFAULT NULL,
  `NewsContent` text,
  `NewsSumaary` text,
  PRIMARY KEY (`idNews`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
--
-- Definition of table `notificationz`
--
DROP TABLE IF EXISTS `notificationz`;
CREATE TABLE `notificationz` (
  `nid` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
  `TID` varchar(10) NOT NULL,

  `Stid` int(10) unsigned NOT NULL,

  `Td` int(10) unsigned DEFAULT NULL,

  `notification` text NOT NULL,

  PRIMARY KEY (`nid`),

  KEY `FK_notificationz_Teacher` (`TID`),

  KEY `FK_notificationz_Student` (`Stid`),

  KEY `FK_notificationz_Topic` (`Td`),

  CONSTRAINT `FK_notificationz_Student` FOREIGN KEY
(`Stid`) REFERENCES `student` (`Stid`) ON DELETE CASCADE
ON UPDATE CASCADE,

  CONSTRAINT `FK_notificationz_Teacher` FOREIGN KEY
(`TID`) REFERENCES `teacher` (`TID`),

  CONSTRAINT `FK_notificationz_Topic` FOREIGN KEY (`Td`)
REFERENCES `topic` (`Td`) ON DELETE CASCADE ON UPDATE
CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Definition of table `program`

--

DROP TABLE IF EXISTS `program`;

CREATE TABLE `program` (

  `Pid` varchar(10) NOT NULL,

  `pname` varchar(145) NOT NULL,

  `duration` int(10) unsigned NOT NULL,

  `NOS` int(10) unsigned NOT NULL,

  `Did` varchar(10) NOT NULL,

  PRIMARY KEY (`Pid`),

  KEY `FK_program_Dept` (`Did`),

  CONSTRAINT `FK_program_Dept` FOREIGN KEY (`Did`)
REFERENCES `dept` (`Did`) ON DELETE NO ACTION ON UPDATE NO
ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Definition of table `semester`
--
DROP TABLE IF EXISTS `semester`;
CREATE TABLE `semester` (
  `SID` int(10) unsigned NOT NULL,
  `sname` varchar(45) NOT NULL,
  PRIMARY KEY (`SID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Definition of table `student`
--
DROP TABLE IF EXISTS `student`;
CREATE TABLE `student` (
  `Stid` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `stdname` varchar(45) NOT NULL,
  `email` varchar(45) NOT NULL,
  `contact` varchar(45) NOT NULL,
  `DOB` date NOT NULL,
  `fname` varchar(45) NOT NULL,
  `mname` varchar(45) NOT NULL,
  `Pid` varchar(10) NOT NULL,
  `RegDate` date DEFAULT NULL,
  `username` varchar(20) DEFAULT NULL,
  `current` int(10) unsigned DEFAULT NULL,
  PRIMARY KEY (`Stid`),
  KEY `FK_Student_program` (`Pid`),
  KEY `FK_student_login` (`username`),
  CONSTRAINT `FK_student_login` FOREIGN KEY (`username`)
REFERENCES `login` (`Username`) ON DELETE CASCADE ON
UPDATE CASCADE,
```

```
  CONSTRAINT `FK_Student_program` FOREIGN KEY (`Pid`)
REFERENCES `program` (`Pid`) ON DELETE NO ACTION ON UPDATE
CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=114305 DEFAULT
CHARSET=latin1;

--

-- Definition of table `teacher`

--

DROP TABLE IF EXISTS `teacher`;

CREATE TABLE `teacher` (

  `TID` varchar(10) NOT NULL,

  `Tname` varchar(45) NOT NULL,

  `Qualification` varchar(145) NOT NULL,

  `contacts` varchar(45) NOT NULL,

  `email` varchar(45) NOT NULL,

  `Username` varchar(20) DEFAULT NULL,

  PRIMARY KEY (`TID`),

  KEY `FK_teacher_login` (`Username`),

  CONSTRAINT `FK_teacher_login` FOREIGN KEY (`Username`)
REFERENCES `login` (`Username`) ON DELETE CASCADE ON
UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Definition of table `topic`

--

DROP TABLE IF EXISTS `topic`;

CREATE TABLE `topic` (

  `Td` int(10) unsigned NOT NULL AUTO_INCREMENT,

  `Tdate` datetime NOT NULL,

  `CID` varchar(10) NOT NULL,

  `TopicTitle` varchar(145) DEFAULT NULL,

  PRIMARY KEY (`Td`),
```

```
  KEY `FK_Topic_Course` (`CID`),

  CONSTRAINT `FK_Topic_Course` FOREIGN KEY (`CID`)
REFERENCES `course` (`CID`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
```

Views

```
--

-- Definition of view `book_course`

--

DROP TABLE IF EXISTS `book_course`;

DROP VIEW IF EXISTS `book_course`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `book_course` AS select `b`.`Bid` AS
`Bid`,`c`.`CID` AS `CID` from (`books` `b` join `course`
`c`) where (`b`.`CID` = `c`.`CID`);

--

-- Definition of view `dept_course`

--

DROP TABLE IF EXISTS `dept_course`;

DROP VIEW IF EXISTS `dept_course`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `dept_course` AS select `c`.`CID` AS
`CID`,`c`.`Cname` AS `Cname`,`c`.`credit` AS
`credit`,`c`.`SID` AS `SID`,`d`.`Did` AS `Did` from
(`course` `c` join `dept` `d`) where (`d`.`Did` =
`c`.`Did`);

--

-- Definition of view `teacher_course`

--

DROP TABLE IF EXISTS `teacher_course`;

DROP VIEW IF EXISTS `teacher_course`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `teacher_course` AS select `t`.`TID`
AS `TID`,`c`.`CID` AS `CID` from (`teacher` `t` join
`course` `c`) where (`c`.`TID` = `t`.`TID`);
```