

**ONLINE STUDENT PROFILE MANAGEMENT SYSTEM**

by

**GREESHMA MALGIREDDY**

B.Tech., Jawaharlal Nehru Technological University, 2007

**A REPORT**

submitted in partial fulfillment of the requirements for the degree

**MASTER OF SCIENCE**

Department of Computing and Information Sciences

College of Engineering

**KANSAS STATE UNIVERSITY**

Manhattan, Kansas.

2010.

Approved by:

Major Professor

Dr. Daniel Andresen.

## **Abstract**

All the students who are enrolled in Computing and Information Sciences (CIS) major in Kansas State University are required to submit their Program of Study (POS) which they manually do by filling in all the necessary details and submit the form to the department.

The main objective of this project is to develop an online submission of program of study. The online student profile management system is a web-based application that provides students of CIS major to submit their program of study in an easy and efficient manner.

This application mainly allows the students to enter their personal information (viz., contact information, previous education) and to choose core as well as non-core courses of their choice. In addition the faculty of CIS department can also log on to the application and view the POS of the students by entering their wildcat ID.

The primary focus is to get familiar with .NET framework and to code in C#.NET. This in turn uses MS SQL server 2005 as database for storing and retrieving of data. This project is implemented using C#.NET on Microsoft visual studio 2005.

# Table of contents

List of Figures .....	v
List of Tables .....	vi
Acknowledgements.....	vii
CHAPTER 1- Introduction .....	1
1.1 Motivation .....	1
1.2 Objective .....	1
1.3 Project Overview.....	2
1.3.1 Study of the system.....	2
1.3.2 Roles of the System .....	2
1.3.3 Inputs and Outputs.....	3
1.3.4 System Architecture .....	4
1.4 Requirement Specification .....	4
1.4.1 Scope .....	4
1.4.2 Goal .....	4
1.4.3 System Requirement Analysis.....	4
1.4.4 Assumptions .....	6
1.4.5 Software Specifications .....	6
CHAPTER 2 – Related Work.....	7
2.1 Tools and Technologies .....	7
2.1.1 ASP.NET .....	7
2.1.2 Visual studio 2005 .....	8
2.1.3 SQL Server 2005 .....	8
2.1.4 AJAX .....	9

CHAPTER 3 - Design and Implementation .....	10
3.1 System Architecture .....	10
3.2 Student Profile Management System Architecture .....	10
3.2.1 Presentation tier .....	12
3.2.2 Business tier.....	12
3.3 Database Diagram .....	18
3.4 Screen Shots .....	20
CHAPTER 4 - Testing.....	29
4.1 Unit Testing.....	29
4.1.1 Tool for Unit testing .....	30
4.1.2 Test cases and Results .....	30
4.2 Performance Testing .....	31
4.2.1 Tool for performance testing .....	32
4.2.2 System Configuration.....	32
4.2.3 Test plan .....	32
4.2.4 Test plans Results and Evaluation .....	33
4.2.4 Testing Analysis .....	38
CHAPTER 5 - Conclusion and Future Work .....	40
5.1 Conclusion.....	40
5.2 Over all Experience.....	40
5.3 Future Enhancements to the project .....	41
REFERENCES .....	42

## List of Figures

Figure 2.1 ASP.NET Framework Architecture [2] .....	8
Figure 3.1 Three tier architecture [6] .....	10
Figure 3.2 Data flow among the three tiers [7] .....	12
Figure 3.3 Student Use case Diagram .....	14
Figure 3.4 Faculty Use case Diagram .....	15
Figure 3.5 Class Diagram .....	16
Figure 3.6 Database diagram of the student profile management system.....	19
Figure 3.7 Screen shot of the home (login) page .....	20
Figure 3.8 Screen shot of the new account registration page.....	21
Figure 3.9 Screen shot of the web page after a new user logins in. ....	22
Figure 3.10 Screen shot of the web page where a student selects program option and courses. ....	23
Figure 3.11 Screen shot of the preview page .....	24
Figure 3.12 Screen shot of the Edit Profile Page. ....	25
Figure 3.13 : Screen shot of the page after Non-Thesis students submits his/her profile. ....	26
Figure 3.14 Screen shot where a faculty member enters student's wildcat ID. ....	27
Figure 3.15 Screen shot of the page where a faculty member views student's profile for the given WID. ....	28
Figure 4.1 Screen Shot of NUnit Test case Execution.....	31
Figure 4.2 Graph results and Summary report for Home page. ....	34
Figure 4.3 Graph results and Summary report for courses selection page.....	36
Figure 4.4 Graph results and Summary report for faculty page.....	38

## List of Tables

Table 4.1 System Configuration .....	32
--------------------------------------	----

## **Acknowledgements**

I would like to express my deepest gratitude to my major professor Dr. Daniel Andresen for his constant help, guidance and encouragement throughout the project.

I would also like to thank Dr. Torben Amtoft and Dr. Mitchell Nielsen for accepting to serve in my committee and for their support during this project.

I finally would like to thank my family members and friends for their constant encouragement and support.

# **Chapter 1 - Introduction**

Generally in all universities filing a Program of Study (POS) by the students is a common phenomenon. In our university, students do this manually by downloading the POS form and submit in the department after filling in all the necessary details and getting it endorsed by either major professor or academic advisor. The proposed system, i.e., the online student profile management system helps students to file their Program of study online. This system provides a simple, easy, convenient and efficient way to file the program of study at any point of time.

## **1.1 Motivation**

The motivation for this project comes from my urge to learn different technologies. I don't have enough experience working on .NET platform until now and through this project I have gained a good learning experience by exposing myself to various technologies like visual studio 2005, SQL Server 2005 for database design, web technologies like AJAX.

## **1.2 Objective**

The main objective of the project is to develop a website which provides students to logon onto the application and submit their program of study in a simple and easy way. The application is implemented using C#.NET programming language on Visual Studio 2005. Another objective is that this application also allows faculty to logon and view the students' program of study by entering their wildcat id. Another objective is to update the student's profile whenever he wishes to edit his information i.e., either the personal or academic information at any point in login session. Yet another objective is to display the total number of credits depending on the courses chosen by the student.



## **1.3 Project Overview**

### **1.3.1 Study of the system**

The website is been designed keeping the students of computing and information sciences department in mind and the main purpose is to provide students an easy, simple and efficient manner of filing the program of study. The user interfaces that are developed in this website help the students with different transactional states like entering the personal information, editing the profile, updating the profile etc., allowing the user to choose from the dropdown menu. The inclusion of this mechanism makes users feel comfortable while using the website.

### **1.3.2 Roles of the System**

There are mainly two important roles in the proposed system.

1. Student
2. Faculty

I shall explain the functionality of the two roles in detail in the below sections.

#### **Student:**

The student plays a major role in this application. If he/she is using the application for the first time, they get started by creating a new account by choosing a username and password. At this point they are also asked to choose the role i.e., either student or faculty. After successful registration the user will see Degreeform page where he will enter his personal information as well as previous education information. After providing all the information the student will be taken to degreeformcourses page where they can choose the program option from thesis and non-thesis and also select courses with at least one from each of the five areas and these form the core courses and also the non-core courses. The student can submit the POS for approval by academic advisor or thesis advisor based on the program option chosen by the student.

The student can any time edit the profile if he/she wishes to consider some other course and update his profile and submit the POS. If he/she is a returning user, then after a successful login the user will view his/her POS.

The courses differ with the selection of program option. In addition to the five core courses, for the non-thesis option there are six non-core courses (including report) and for thesis option there are four non-core courses along with the details about the thesis topic viz., thesis title, names of committee members etc.,

#### Faculty:

Faculty is another user of the system. A new faculty member will register through the create account and after this is done, he/she will enter the wildcat ID of the student which fetches the details of the POS of the student. After verification he/she then approves it by endorsing it. If he is a returning user then he also does the same action by directly getting redirected to the page where he can enter Wildcat ID of the student.

### **1.3.3 Inputs and Outputs**

The main inputs, outputs and major functions of the system are as follows:-

#### Inputs from student role:

- The student enters his/ her username and password.
- The student enters his personal information.
- The student chooses program option.
- The student selects various courses.
- He/she can edit the details and so on.

#### Inputs from faculty role:

- The faculty enters his/her username and password

- He/she enters the wildcat ID of the student.

Expected Outputs:

- The student views his/her profile.
- The faculty views the student's profile.

### **1.3.4 System Architecture**

The architecture of the current application uses a layered architecture where its functionality can be divided into three different layers thus; rendering the architecture to be the three tiered. This three-tier architecture has three layers within it. They are the presentation layer, business layer and the data layer. The architecture is more explained in detail in the future sections.

## **1.4 Requirement Specification**

### **1.4.1 Scope**

The scope of the current application, “online student profile management system” is that it ensures the master's students in CIS major to file their program of study in an easy and hassle-free way. Thereby it reduces the amount of paper work related to it, which is in effect now.

### **1.4.2 Goal**

The main goal of implementing this project is to get hands on experience working on the technologies used in this project which are ASP.NET, C# programming language and the SQL server management studio. By obtaining knowledge of coding in C#.Net and designing efficient databases in SQL server will help me a lot in future projects involving these technologies.

### **1.4.3 System Requirement Analysis**

The most important goal of the application as mentioned above is to provide students an easy, efficient and interactive interface way to file their POS online. I have carefully worked on understanding what students exactly need and how to make it convenient so that it will be easy for them to use.

As mentioned above in the roles, there are two super users of this application. One is the student and the other is the faculty member. Presently I have considered only these two roles. A new user of the application i.e., a student should be able to create a new account by choosing his/her username, password. The username should be a minimum of eight characters and password can be anything. He is also supposed to choose the role like the faculty or the student and then he should be able to create it. After this process is done, he should be able to enter in his personal information on the next page. Personal information here refers to Name (which is first name, middle name and last name), student ID, type of address (permanent, local or mailing), current address including street number, state, country (USA) with zip code.

In addition to this, the user should be able to provide his previous education history with name of previous institution where he studied and the graduation year. If he wishes to enter more than one institution he should be able to do that. He should be able to fill in maximum of three previous education institutions. After successfully entering all this information, he should be able to choose the courses. Before proceeding to choose courses, he should be able to select the program option from the drop down menu. And the display should change with selection of the program option. That is it should be different for both thesis and non-thesis options. After choosing this, he should be able to choose one course from the five drop down menus provided which form the core courses. If it is non-thesis option, five drop down menus with courses in them and six other drop downs for non-thesis (including the one for report) should be displayed. If it is a thesis option then five drop down menus and four drop down menus for non-core courses and information about thesis report like, thesis advisor, committee members and the thesis topic are to be displayed. While this process is going on even the total of credits for all these courses is needed to be calculated and is to be displayed. After done with this process finally he can submit his profile and again this depends on his program option, if his option is non-thesis he would be submitting to graduate advisor and if it is thesis he would submit to thesis advisor.

After choosing all the necessary course information he should be able to redirect to page where he could see a preview of his profile and they should be able to edit the profile whenever they want to.

If he is a returning user then he should be able to see the same and edit his profile. When the user is a faculty member, if he is a new user then he should be able to do the same process of choosing user name, password and role. After successfully setting up an account he should be

able to enter the wildcat ID of the student and should be able to see the profile of that particular student and after checking the information, he should sign for approval and submit it to graduate school.

The application can be made more interactive by using certain help tools so that the users will have an idea about how to proceed when they arrive on courses selection page. In addition to this, I have done research on various tools to building this application and I used certain stronger features while designing this application. ASP.NET controls and AJAX toolkit controls are the tools and controls which I used in order to improve the interactivity, navigation and usability.

#### **1.4.4 Assumptions**

The following assumptions are made when the users want to use this application:-

- The user will have access to internet.
- He will have Mozilla fire fox while using the student profile management system.
- The faculty member will enter a valid student ID.
- Students will enter valid information while creating their profile.

#### **1.4.5 Software Specifications**

- The development environment of online student profile management system will be Microsoft Visual studio .NET.
- The proposed application will be written in C#.NET programming language.
- The application will be tested on Windows vista platform.

The database that will be used in this application for storing and retrieving data will be SQL Server 2005.

## **Chapter 2 – Related Work**

### **2.1 Tools and Technologies**

This chapter discusses in detail about the tools and technologies used in this project. The tools and technologies used in this application are ASP.NET 2.0, Microsoft visual studio 2005 and MS SQL server 2005.

#### **2.1.1 ASP.NET**

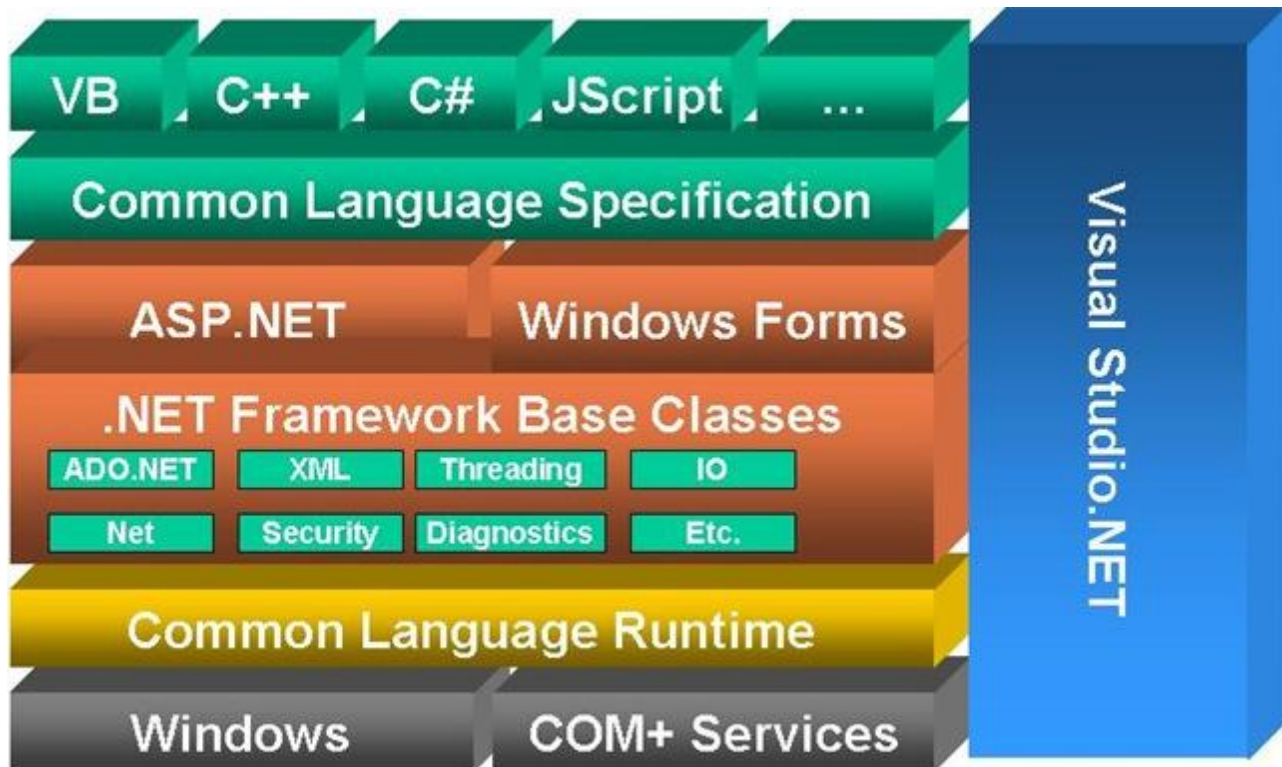
ASP.NET is a part of .NET framework and is a web-application framework. It is built on the Common Language Runtime (CLR) which allows programmers to build dynamic web applications, web services and web sites and also allows them to write ASP.NET code using any .NET language. The languages that are compatible with CLR include Microsoft Visual Basic, C#, J# and Jscript.NET. These languages enable the programmers to develop the applications that benefit from the CLR, inheritance, type safety etc,

ASP.NET consists of several other components like page and controls framework. This runs on a web server to produce and render ASP.NET web pages dynamically. This in turn also enables programmers to encapsulate common UI functionality in easy to use reusable controls. By providing various themes and skins it gives various features to control the overall look and feel of the website.

The ASP.NET compiler compiles all the application components including pages and controls into an assembly so that ASP.NET environment can be used to service requests from the users. It has several other features like debugging support, health monitoring and performance features, xml web services framework, an extensible designer and hosting environment and application life cycle management.

The architecture of ASP.NET is based on the CLR which runs as virtual machine on operating system. This CLR is layered on top of the windows and COM+ services. On top of it are the .NET framework base classes which provide classes that can be called from any .NET language. These contain ADO.NET, XML, IO, security, threading etc., ADO.NET which is a set of classes provide data access support for the .NET framework based on ADO. On the top of .NET framework base classes are ASP.NET and windows forms. ASP.NET offers web forms and web services where as the windows forms are used by traditional client applications (windows applications). Common Language Specification (CLS) is on top of these two which makes sure that each language has common set of features. Layered on top of CLS are various .NET programming languages like VB, C++, C#, J#, Jscript etc., [1]

Figure 2.1 ASP.NET Framework Architecture [2]



### 2.1.2 Visual studio 2005

Visual studio is an Integrated Development Environment (IDE) developed by Microsoft and it helps to build various ASP.NET web applications, XML Web services by providing various development tools. It also helps in building console as well as GUI applications that can be in native code combined with managed code for all platforms that can be supported by Windows, .NET framework etc., There are several other built-in tools which include a forms designer which can also be used to build GUI applications, web/ class/ database schema designers etc.,

Visual studio allows any programming languages that are supported by the code editor and debugger. Several built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic.NET) and C# (via Visual C++). It also supports XML, HTML, XHTML, JavaScript and CSS. [3]

### 2.1.3 SQL Server 2005

Microsoft SQL server is a relational model database server produced by Microsoft. This is the successor to SQL server 2000 and comprises of T-SQL and ANSI SQL as its query languages. This provides tools that developers need to build new classes of database applications.

In order to integrate with .NET framework, SQL CLR was introduced with SQL Server 2005. Using this integration, programmers can code stored procedures, functions and triggers in .NET framework language of their choice. VB.NET and C# languages offer object-oriented constructs, namespaces, classes and arrays. Many tasks which are difficult to perform in Transact-SQL can be accomplished by managed code. The integration between CLR and SQL server provides many benefits such as enhanced safety and security, common development environment, performance, scalability and enhanced programming model. This also supports web services in which one can develop XML web services in the database tier which makes SQL server an HTTP listener.

The current application utilizes all these features of SQL server to store and retrieve the data of the student's profile. [4]

#### **2.1.4 AJAX**

AJAX is shorthand for Asynchronous JavaScript and XML. This is a group of inter related web development techniques used on the client side to create interactive web applications. Web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page using AJAX. It uses a combination of HTML and CSS for styling and marking up information and to dynamically display and interact with the information presented. In order to render the web application and handle requests from customers, Ajax engine works within the web browser through JavaScript and the DOM. The benefit of this is that when the Ajax engine is handling requests, it allows the interaction with the application and user to happen asynchronously and also independent of the interaction with the server while the engine is holding most of the information in itself. [5]

Since it has all these features, Ajax is meant to increase the web page's interactivity, speed and usability. In this current application, I have used Ajax to display the change in selection of courses, when the student selects the program option as either Thesis or Non-thesis.

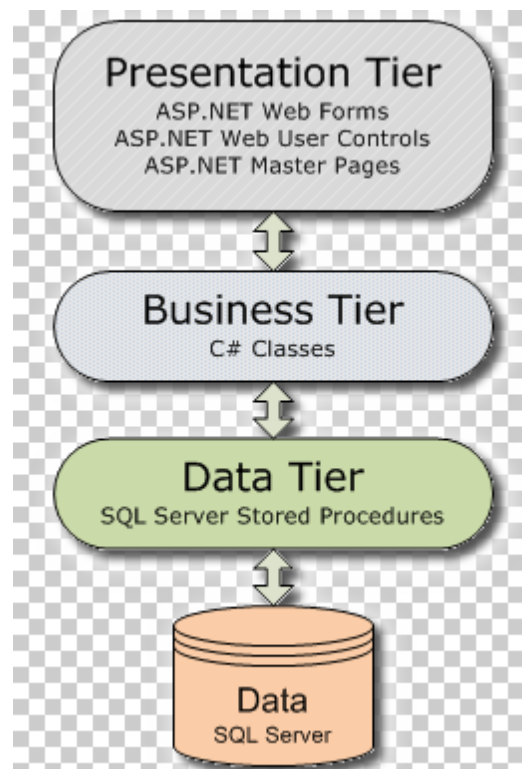


## Chapter 3 - Design and Implementation

### 3.1 System Architecture

This chapter describes about the architecture used in this project and the three tier architecture is used. The three tier architecture is a client-server architecture in which user interfaces, functional business logic, data storage and data access are developed and maintained as independent modules, often on different platforms. This model is considered to be software architecture and a software design pattern. This architecture allows any of the three tiers to be upgraded independently as requirements or technology change and is described in detail in the below sections.

Figure 3.1 Three tier architecture [6]



### 3.2 Student Profile Management System Architecture

The online student profile management system architecture as mentioned above uses the three tier architecture. From the figure 2.1, it can be noticed that the architecture comprises of presentation, business and data tiers. These three tiers can also be referred to as logical layers where each layer performs a specific function and forms the reusable section of the code. These layers interact with each other layers in order to perform certain tasks. The presentation layer

which is also called as application layer provides various forms web forms, user controls, master pages etc., The presentation tier of the current application provide this functionality for the programmers to design the website. It has got ASP.NET web forms, web controls, master pages, user controls, Ajax etc., for the above said purpose. In addition to this programmers also use this layer to get or set the data back and forth. [6]

The ASP.NET 2.0 has the support for all the master pages which has all the controls mentioned above and the user can make use of them for designing web pages and only the content tab will be loaded for different pages. Hence, I chose Visual studio .Net 2005 to design and code the ASP.NET pages to make the website more interactive and also attractive. As it has got support for the master pages too, I made use of this feature in order to make the website pages look uniform and pretty as it has got common controls on the web pages making it easier for the programmer to design. This is the layer where the user interacts with the application part. The user provides input through this layer. From this layer further processing takes place by supplying the information to the layer below it which is the business layer or the middle layer.

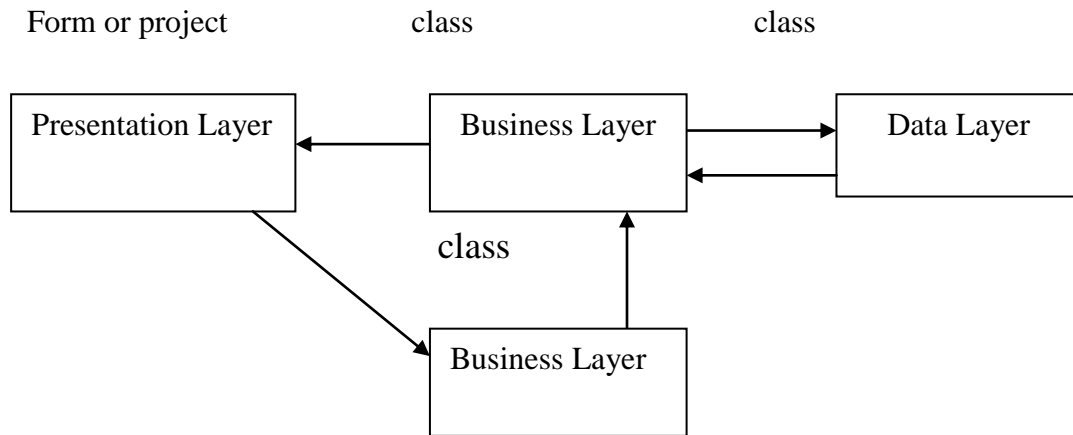
The business layer which is also called as middle layer is responsible for sharing and controlling the business logic of the application. Programmers write the functions in this layer for getting data from the application layer or the presentation layer and passes through the data access layer. This contains several C# classes and this can be noticed from the figure shown above. In addition to the above said functionalities, this tier is also responsible for increasing the code transparency and also for altering the database. The classes of this application viz., course, pos, user, login info, thesis advisor, student, faculty, graduate advisor are contained in this layer only. The business layer can be more clearly understood by the use case and class diagrams of the application which are shown below.

The current application student profile management system uses SQL server 2005 and its database. This SQL server makes the search effective and convenient whenever the user (student or the faculty member) requests for the POS. The database schema for this application is shown in figure 3.4 which comprises of five tables viz., student, login detail, courses, student courses and thesis student courses. From the figure all the fields and keys of the tables can be noticed.

So because of all the above said functionalities of the three tier architecture, I felt it matches with all my requirements of the project and hence I have decided to go with three tier architecture for my current application.

The figure 3.2 which is below describes the data flow among the three tiers.

Figure 3.2 Data flow among the three tiers [7]



The following sections describes in detail about the above said three tiers.

### 3.2.1 Presentation tier

This is the top most level of the application and this is the user interface. The main function of the interface is to translate the tasks and results to something which can be understood by the user. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network. In this presentation layer the programmers design the forms using the controls like textbox, labels, command buttons etc., In the above diagram we have ASP.NET web forms, web user controls and ASP.NET master pages. In the current application this tier consists of ASP.NET web forms which contain the ASP web controls, user controls and AJAX controls. I have used Visual studio .NET 2005 version to design and code the ASP.NET pages.

### 3.2.2 Business tier

This layer is pulled out from the presentation tier and it controls an application's functionality by performing detailed processing. This layer is the class where the programmers write the functions which get the data from the above layer (presentation layer) and passes through the data access layer. The presentation layer communicates with the data access layer through this layer only. This layer validates the input and ensures that the input is correct before proceeding further and often ensures that output is correct. The business logic lies within this layer and this makes the logic reusable in all the applications. The use case and class diagrams of this application are described in detail in the below sections:-

### 3.2.2.1 Use Case Diagrams

In the Unified Modeling Language (UML), the use case diagram is a type of behavioral diagram defined by and created from a use-case analysis. It represents a graphical over view of the functionality of the system in terms of actors, which are persons, organizations or external system that plays a role in one or more interaction with the system. These are drawn as stick figures. The goals of these actors are represented as use cases, which describe a sequence of actions that provide something of measurable value to an actor and any dependencies between those use cases.

The associations between actors and use cases are indicated in use case diagrams by solid lines and this exists whenever an actor is involved with an interaction described by use case. [8]

#### Student Use case:

The student can create an account, login, Update his/her personal information, select program option (either thesis or non-thesis), select courses (either core or non-core), edit his profile, view his profile and logout.

#### Faculty Use case:

The faculty member can create an account, login, enter student's ID, view student's profile, approve pos and logout. The following figures show the use case diagrams of the roles viz., student and Faculty of the student profile management system.

Figure 3.3 Student Use case Diagram

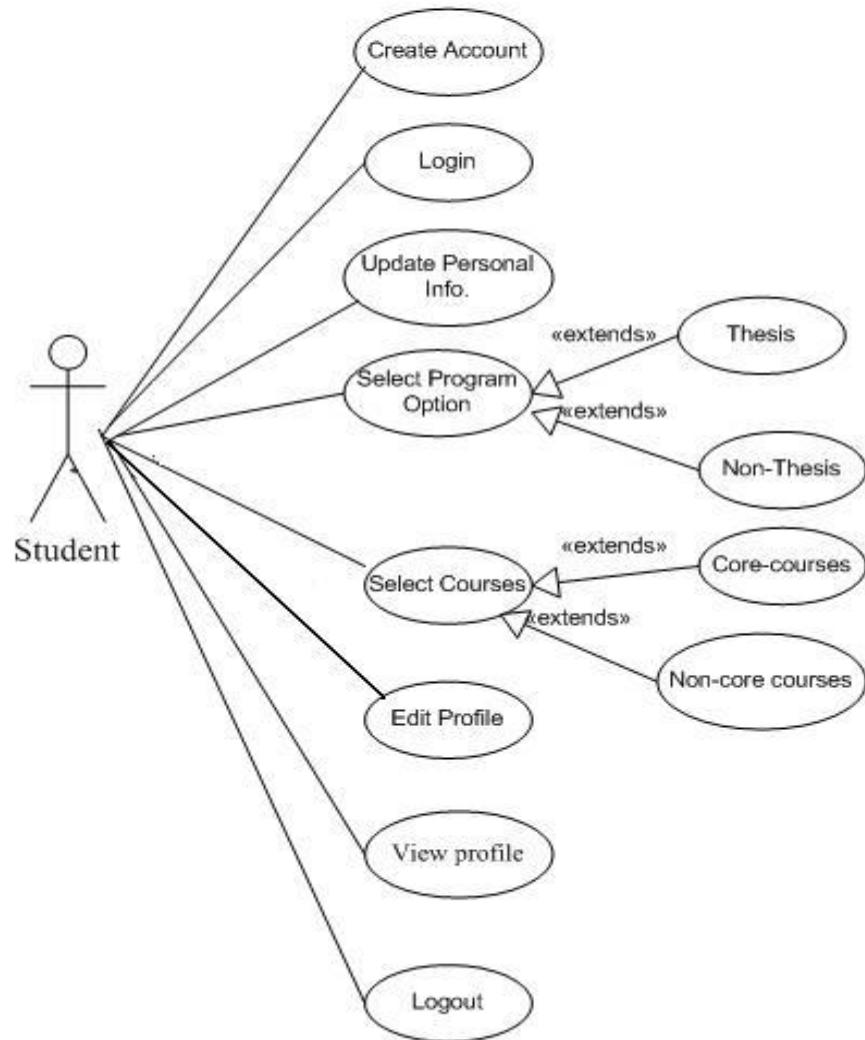
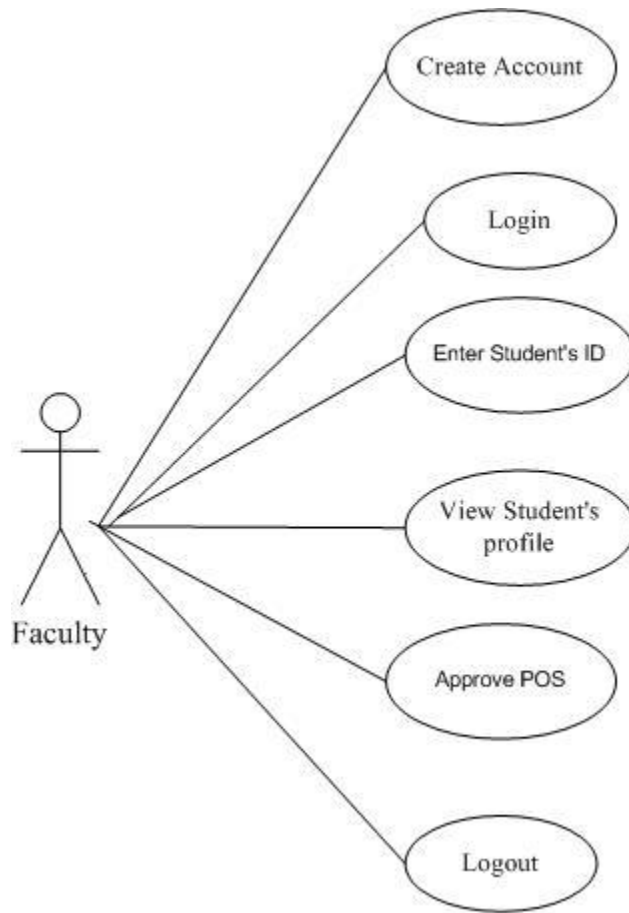


Figure 3.4 Faculty Use case Diagram

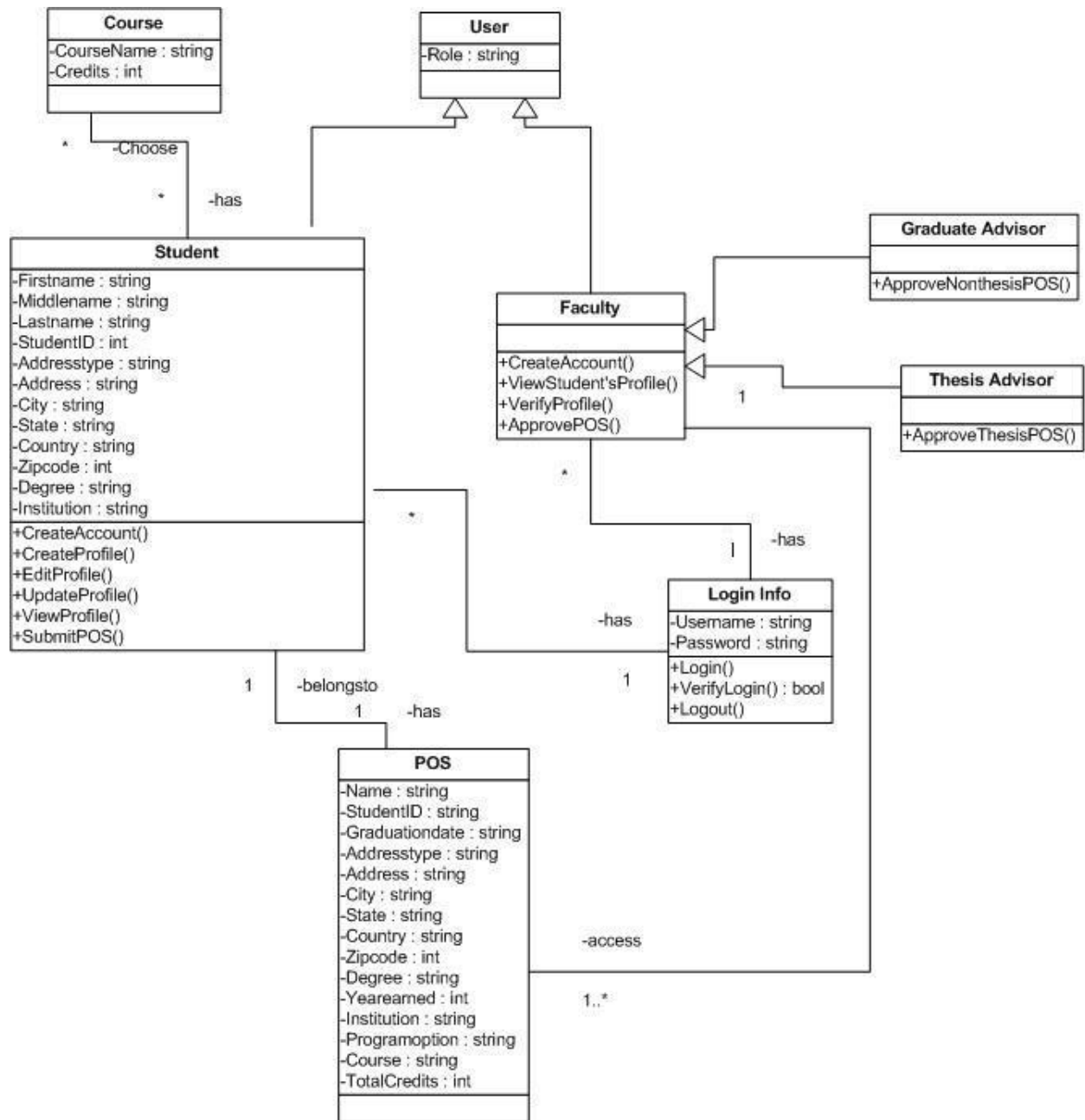


### 3.2.2.2 Class Diagrams

In the UML, class diagram is a type of static structure diagram which describes the structure of a system by showing the classes, attributes and the relationships between the classes of a system. It is the main building block in object oriented modeling. The classes represent both the main objects and or interactions in the application. The class diagram consists of classes which are represented in boxes which contain three parts. The name of the class is contained in upper part, with the attributes of classes in middle part and the bottom part contains the methods or operations that the classes undertake. With detailed modeling, the classes of the conceptual design are split into number of subclasses. [9]

The following figure shows the class diagram of the student profile management system.

Figure 3.5 Class Diagram



The description for each of the classes of the class diagram shown above is given below:-

Student class:

The student class is the basic class of this application and it has got the attributes of First name, middle name, last name, Student ID, address, degree, institution etc. that describes about the basic information of the user. The methods in this class include creating an account, creating a profile, editing the profile, updating the profile, viewing the profile and submitting the profile. One student at a time can submit only one profile. And in this application, one user can view only his profile.

This is associated with several other classes like course, user, login info and POS. With class course it has many-many relationship as there can be more than 1 student in class and student can enroll in more than 1 course. With POS class it is one-one, as student can have only one POS and POS in turn belong to only 1 student. And student is a kind of user class. It has login info and at an instance one student can have only one login info.

#### POS class:

POS class has got various attributes like name, student ID, address, degree, program option, total credits etc., that describes about the profile of a student. It has got association with student and faculty classes. The student and faculty classes have one to many relationships because a faculty member can have access to more than 1 student's POS. This forms another important class of this current application.

#### Faculty class:

The faculty class has no attributes and it has methods like creating account, viewing student's profile, verify profile and approve POS. It has association with several classes like Login info, POS, graduate advisor, thesis advisor and User. A faculty can have only one login information but a login info class can have logins of many faculty members. A faculty can be a graduate or thesis advisor. A faculty has one to many relationships with POS class and it is type of user class.

#### Course class:

The course class has attributes like course name and credits. These credits are for that particular course. And it is associated with the class student. It has many-many relationship with student class as described above.

#### User Class:

The user class has got attributes like role and is associated with Student and faculty classes. They both are type of a User.

#### Login Info:

The login info class has attributes like username and password. And it has got methods like login, logout and verifies login information supplied by the user. It has association with Faculty



and Student classes where a student / faculty member can have only one login information where as login info has information of users of the application.

#### Graduate Advisor:

This class has no attributes but has got one method like Approve non thesis POS, where a graduate advisor approves POS of non-thesis students. It associates itself with Faculty class, where a graduate advisor is type of faculty

#### Thesis Advisor:

Similar to the graduate advisor class, this class has no attributes but has got one method like Approve thesis POS, where a thesis advisor approves POS of thesis students. It associates itself with Faculty class, where a thesis advisor is type of faculty.

### **3.2.2.3 Data tier**

This tier consists of database servers and it is here where the information is stored and retrieved. The Data tier keeps data neutral and independent from application servers or business logic. This is also the class which gets the data from the business tier and sends it to the database or gets the data from the database and sends it to the business layer. In this project I used MS SQL Server 2005 as database and this layer makes use of this database.

### **3.3 Database Diagram**

The database schema of the project has got five tables in it. They are student table, courses table, student courses table, thesis student courses table and login detail table. Description about each of the tables is been given below in detail:-

#### Login detail:

The login detail table consists of all the login information of the users that is of students and faculty members. With fields like user ID, password, role and degree type with user ID being the primary key.

#### Courses:

This table consists of names of all the courses and credits each course carries.

#### Student:

This table has information about the profile of a student with all personal information as well as previous education information. Student ID serves as primary key in this table.

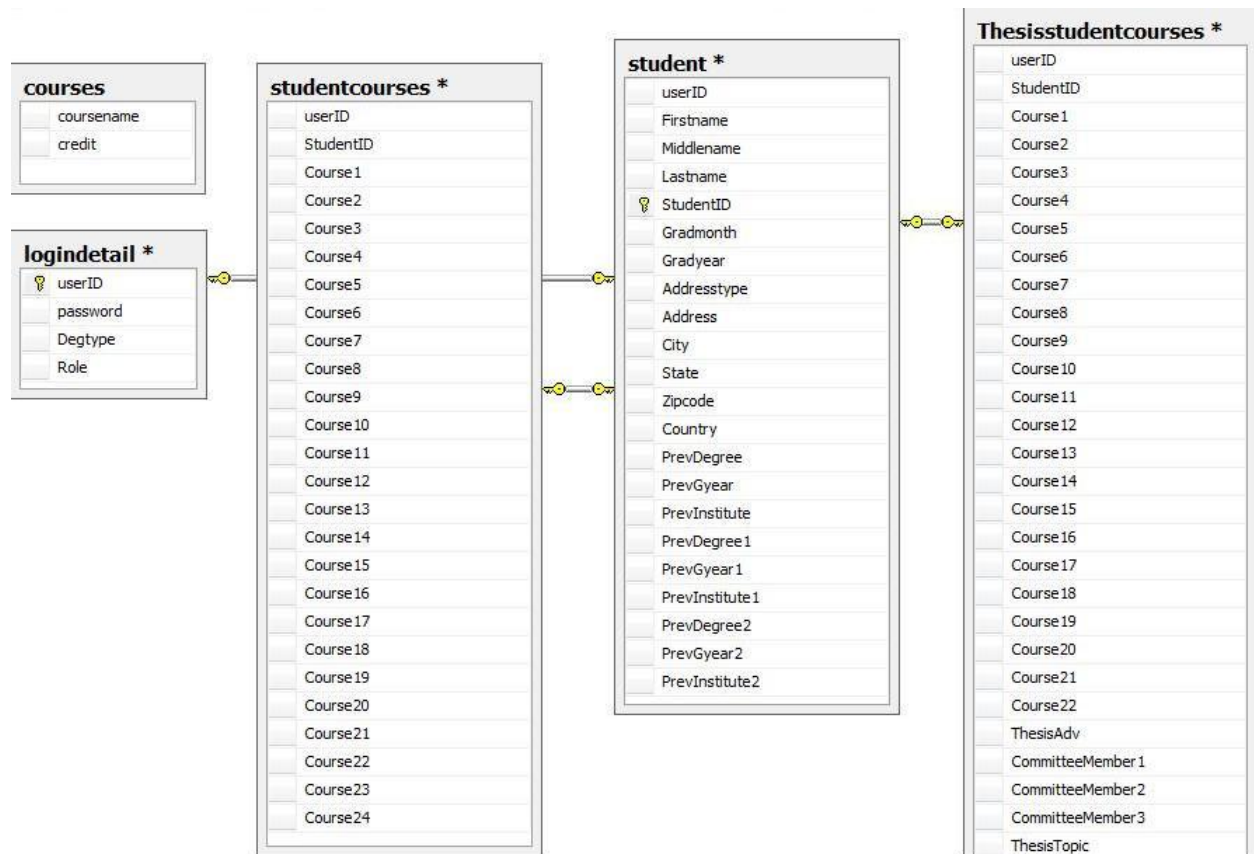
#### Student Courses:

This table has fields of user ID, student ID and the courses non thesis students select. It has student ID as foreign key.

Thesis Student Courses:

This table which is similar to student courses table has fields of user ID, student ID and the courses thesis students select. It has student ID as foreign key.

Figure 3.6 Database diagram of the student profile management system



### 3.4 Screen Shots

Figure 3.7 Screen shot of the home (login) page

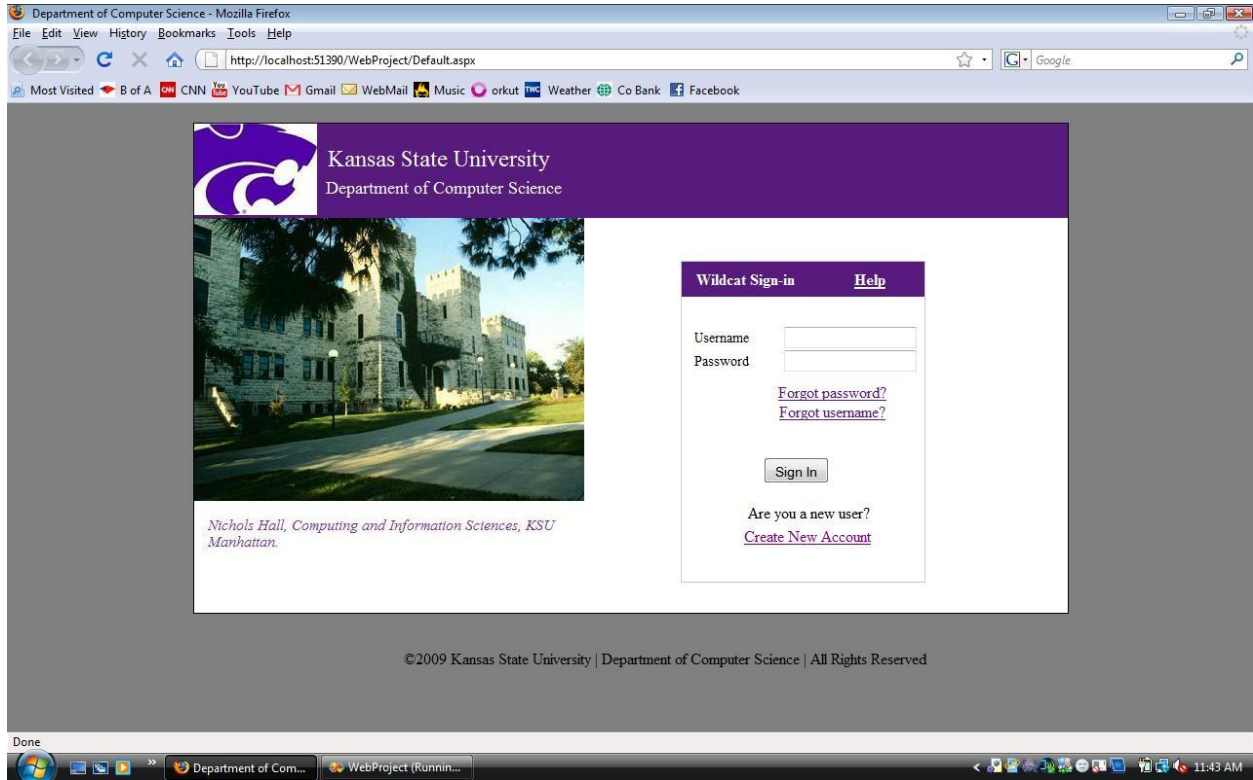


Figure 3.8 Screen shot of the new account registration page

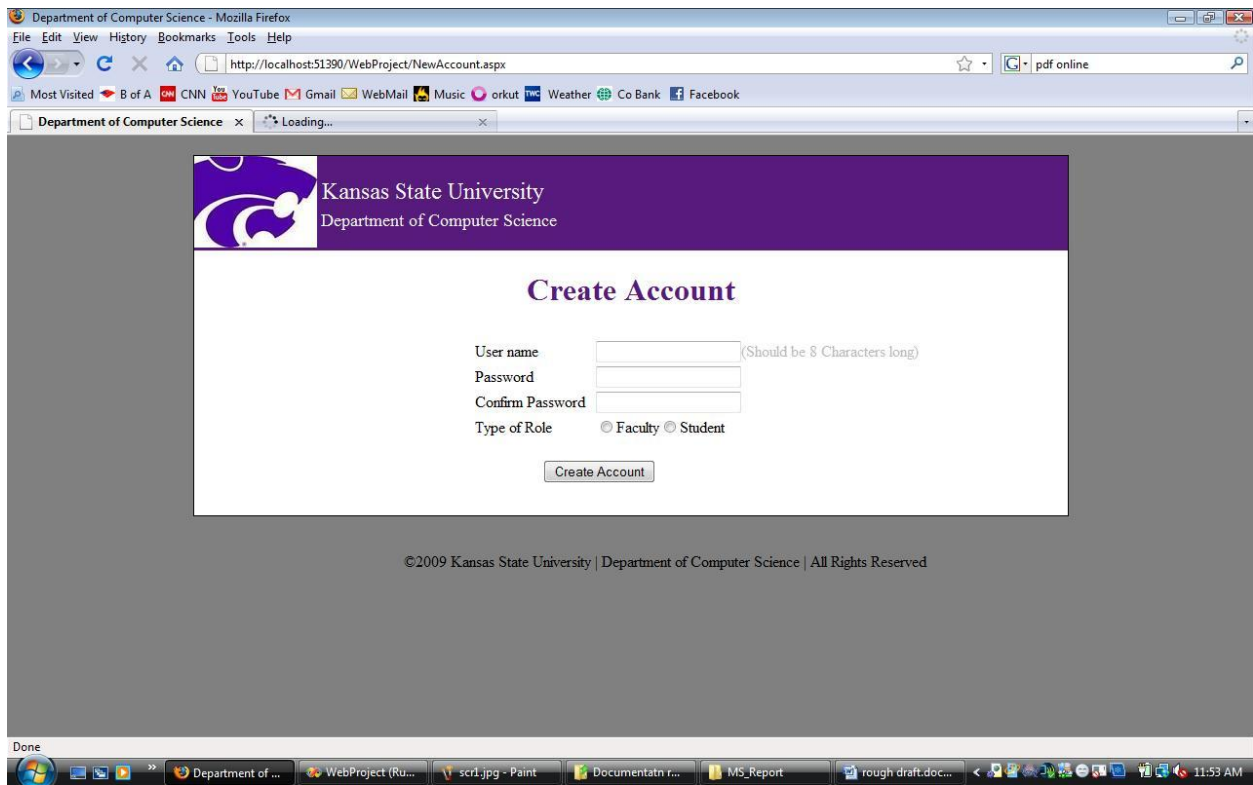


Figure 3.9 Screen shot of the web page after a new user logs in.

Kansas State University  
Department of Computer Science

Welcome account3 | [Logout](#)

**MS CIS DEGREE PLAN (THESIS AND NON-THESIS OPTIONS)**  
Complete and submit during the second semester of graduate study with all courses specified even if you have not taken them yet.

**PERSONAL INFORMATION**

Name: John   Fuller  (eg: "Hans" "John" "Solo")  
Wildcat ID: 2178  (eg: xxxxxxxx)  
Expected Graduation Date: August  2010  (eg: "August" "2009")  
Official Mailing Address: Permanent   
Street Address: 2030 Anderson Avenue   
(eg: "1010 Vattier st")  
City: Manhattan  (eg: "Manhattan") State: Kansas  (eg: "Kansas")  
ZIP Code: 66505  (eg: "66502") Country: USA  (eg: "USA")

**PREVIOUS DEGREE(S) EARNED**

Degree: Bachelor's   
Year: 2006   
Institution: Florida State University

Degree: -Select Degree-   
Year: -Select Year-   
Institution:

Degree: -Select Degree-   
Year: -Select Year-   
Institution:

Figure 3.10 Screen shot of the web page where a student selects program option and courses.

The screenshot displays the 'MS CIS DEGREE PLAN (THESIS AND NON-THESIS OPTIONS)' page. At the top, it features the Kansas State University logo and the Department of Computer Science name. A user is logged in as 'account3'. The page is divided into sections for 'CORE COURSES' and 'OTHER COURSES'. Under 'CORE COURSES', five required courses are listed with dropdown menus and 'Help?' links: IMPLEMENTATION (CIS 736 - Computer Graphics), STRUCTURES (CIS 740 - Software Engineering), LANGUAGES (CIS 771 - Specification and Verifierator), THEORY (CIS 775 - Analysis of Algorithms), and SYSTEMS (CIS 720 - Operating Systems). The 'OTHER COURSES' section lists four additional courses: Course 1 (CIS 721 - Real-Time Systems), Course 2 (CIS 761 - Database Systems), Course 3 (CIS 722 - Operating System Practices), and Course 4 (CIS 825 - Top/Distributed System). The 'THESIS DETAILS' section includes fields for Thesis Advisor (Daniel Andresen), Committee Members (Torben Amtoft and Mitchell Nielsen), Thesis Topic or Title (Data Mining), and Total Credits (33/33). Navigation buttons for '<< Previous' and 'Next >>' are located at the bottom.

Kansas State University  
Department of Computer Science

Welcome account3 | [Logout](#)

**MS CIS DEGREE PLAN (THESIS AND NON-THESIS OPTIONS)**  
Complete and submit during the second semester of graduate study with all courses specified even if you have not taken them yet.

Program Option : Thesis

**CORE COURSES**

IMPLEMENTATION (1 is required): CIS 736 - Computer Graphics [Help?](#)

STRUCTURES (1 is required): CIS 740 - Software Engineering [Help?](#)

LANGUAGES (1 is required): CIS 771 - Specification and Verifierator [Help?](#)

THEORY (1 is required): CIS 775 - Analysis of Algorithms [Help?](#)

SYSTEMS (1 is required): CIS 720 - Operating Systems [Help?](#)

**OTHER COURSES**

Course 1: CIS 721 - Real-Time Systems

Course 2: CIS 761 - Database Systems

Course 3: CIS 722 - Operating System Practices

Course 4: CIS 825 - Top/Distributed System

**THESIS DETAILS**

Thesis Advisor : Daniel Andresen Committee Member : Torben Amtoft

Committee Member : William Hsu Committee Member : Mitchell Nielsen

Thesis Topic or Title : Data Mining

Total Credits: 33/33

<< Previous Next >>

Figure 3.11 Screen shot of the preview page

 **Kansas State University**  
Department of Computer Science

Welcome account7 | [Logout](#)

---

**MS CIS DEGREE PLAN - NON-THESIS OPTION**

---

**PERSONAL INFORMATION**

Name: abc defgh Wildcat ID: 9002  
Expected Graduation Date: February 2010  
Official Mailing Address: Permanent  
Street Address: 1010 kearney  
City: manhattan ZIP Code: 66502  
State: Kansas Country: USA

---

**PREVIOUS DEGREE(S) EARNED**

Degree: Bachelor's	Year: 2007	Institution: OSU
Degree: 0	Year:	Institution:
Degree: 0	Year:	Institution:

---

**IMPLEMENTATION CORE COURSES**

1. CIS 706 - Translator design

---

**STRUCTURES CORE COURSES**

1. CIS 740 - Software Engineering

---

**LANGUAGES CORE COURSES**

1. CIS 705 - Programming Languages

---

**THEORY CORE COURSES**

1. CIS 775 - Analysis of Algorithms

---

**SYSTEM CORE COURSES**

1. CIS 725 - Advanced Computer Networks

---


**OTHER COURSES**

1. CIS 748 - Software Project Management	2. CIS 721 - Real-Time Systems
3. CIS 761 - Database Systems	4. CIS 771 - Specification and Verification
5. CIS 898 - Ms Report	6. CIS 897 - Sem/Grad Computer Science

TOTAL CREDITS

©2009 Kansas State University | Department of Computer Science | All Rights Reserved

Figure 3.12 Screen shot of the Edit Profile Page.



**Kansas State University**  
Department of Computer Science

Welcome account7 | [Logout](#)

---

MS CIS DEGREE PLAN - NON-THESIS OPTION

---

PERSONAL INFORMATION

**Name:**   **Wildcat ID:**

**Expected Graduation Date:**

**Official Mailing Address:**

**Street Address:**

**City:**  **ZIP Code:**

**State:**  **Country:**

---

PREVIOUS DEGREE(S) EARNED

<b>Degree:</b> Bachelor's	<b>Year:</b> 2007	<b>Institution:</b> OSU
<b>Degree:</b> 0	<b>Year:</b>	<b>Institution:</b>
<b>Degree:</b> 0	<b>Year:</b>	<b>Institution:</b>

---

CORE COURSES

**IMPLEMENTATION (1 is required):**  [Help?](#)

**STRUCTURES (1 is required):**  [Help?](#)

**LANGUAGES (1 is required):**  [Help?](#)

**THEORY (1 is required):**  [Help?](#)

**SYSTEMS (1 is required):**  [Help?](#)

---

OTHER COURSES

<p>1. <input type="text" value="CIS 748 - Software Project Management"/></p> <p>3. <input type="text" value="CIS 761 - Database Systems"/></p> <p>5. <input type="text" value="CIS 898 - Ms Report"/></p>	<p>2. <input type="text" value="CIS 721 - Real-Time Systems"/></p> <p>4. <input type="text" value="CIS 771 - Specification and Verification"/></p> <p>6. <input type="text" value="CIS 897 - Sem/Grad Computer Science"/></p>
---	---

[View the list of courses](#)

**TOTAL CREDITS**

©2009 Kansas State University | Department of Computer Science | All Rights Reserved



Figure 3.13 : Screen shot of the page after Non-Thesis students submits his/her profile.

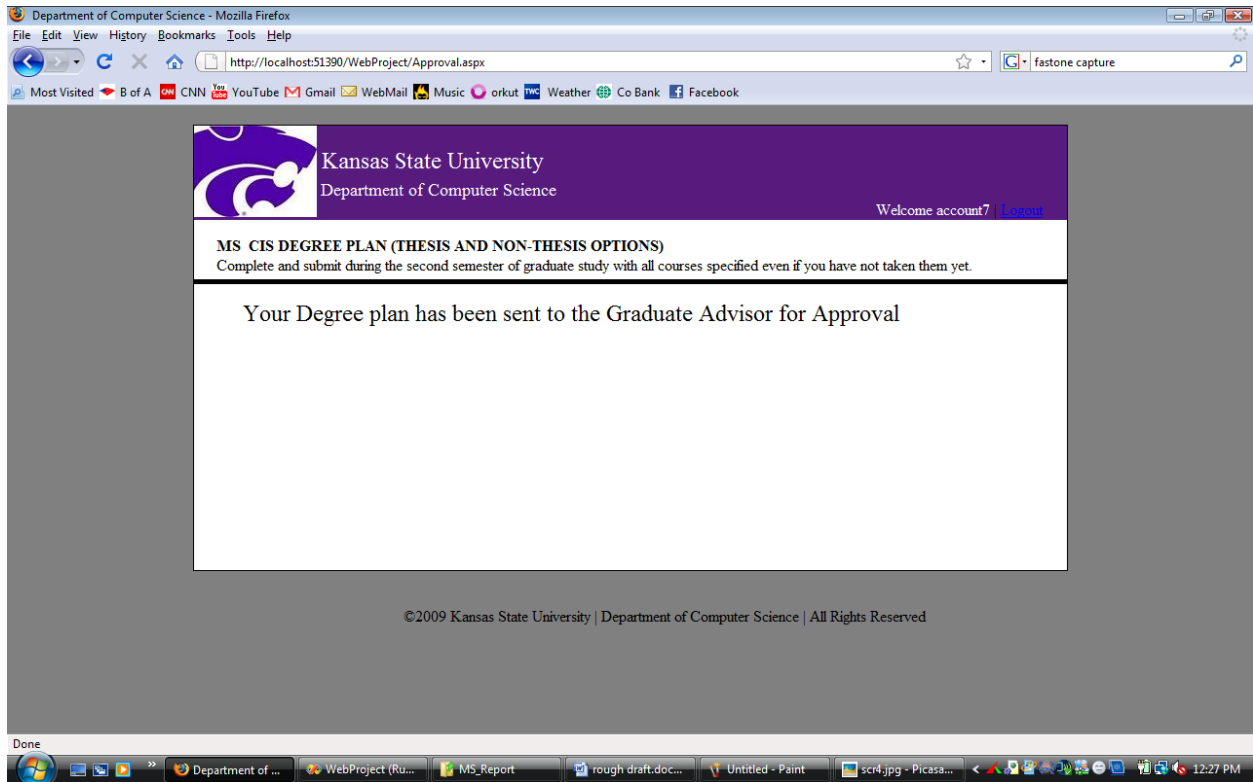


Figure 3.14 Screen shot where a faculty member enters student's wildcat ID.

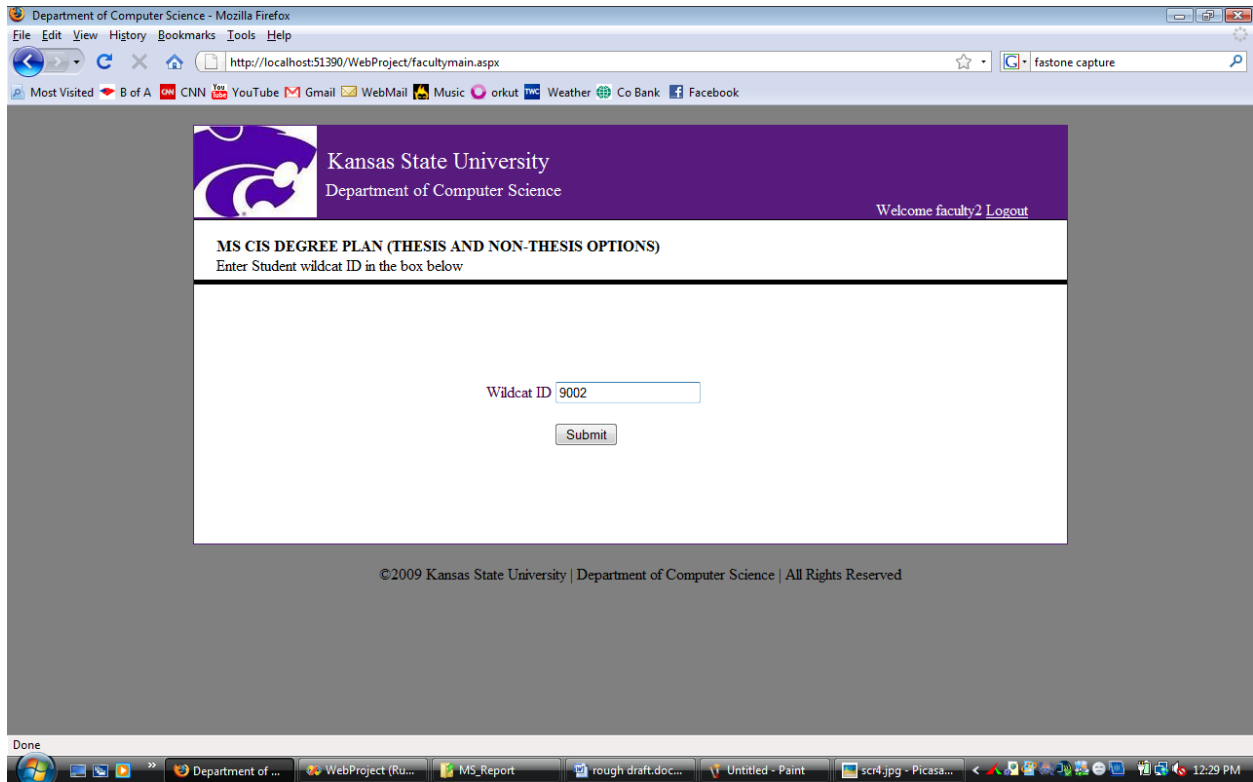



Figure 3.15 Screen shot of the page where a faculty member views student's profile for the given WID.


Kansas State University  
Department of Computer Science
Welcome faculty2 | [Logout](#)

---

MS CIS DEGREE PLAN - NON-THESIS OPTION

PERSONAL INFORMATION

<b>Name:</b> abc defgh	<b>Wildcat ID:</b> 9002
<b>Expected Graduation Date:</b> February 2010	
<b>Official Mailing Address:</b> Permanent	
<b>Street Address:</b> 1010 kearney	
<b>City:</b> manhattan	<b>ZIP Code:</b> 66502
<b>State:</b> Kansas	<b>Country:</b> USA

PREVIOUS DEGREE(S) EARNED

<b>Degree:</b> Bachelor's	<b>Year:</b> 2007	<b>Institution:</b> OSU
<b>Degree:</b> 0	<b>Year:</b>	<b>Institution:</b>
<b>Degree:</b> 0	<b>Year:</b>	<b>Institution:</b>

IMPLEMENTATION CORE COURSES

1. CIS 706 - Translator design

STRUCTURES CORE COURSES

1. CIS 740 - Software Engineering

LANGUAGES CORE COURSES

1. CIS 705 - Programming Languages

THEORY CORE COURSES

1. CIS 775 - Analysis of Algorithms

SYSTEM CORE COURSES

1. CIS 725 - Advanced Computer Networks

OTHER COURSES

1. CIS 748 - Software Project Management	2. CIS 721 - Real-Time Systems
3. CIS 761 - Database Systems	4. CIS 771 - Specification and Verification
5. CIS 898 - Ms Report	6. CIS 897 - Sem/Grad Computer Science

**TOTAL CREDITS**

MAJOR PROFESSOR SIGNATURE

<b>Major Professor :</b>	<b>Date:</b>
--------------------------	--------------

©2009 Kansas State University | Department of Computer Science | All Rights Reserved

## **Chapter 4 - Testing**

Software testing is the process of executing a program with an intension of finding errors. This can also be stated as the process of validating and verifying that a software program is working as expected and is meeting the technical as well as business requirements. This can be implemented at any time in the development process. There are many types of testing software viz., unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc., [10]

I have conducted unit as well as performance testing on my application and these two types of testing are described below:-

### **4.1 Unit Testing**

Unit testing refers to those tests that verify the functionality of a specific section of the code, usually at the function level. In an object oriented environment this is usually at the class level and the unit tests include the constructors and the destructors. The primary goal is to take the smallest piece of testable software, isolate it from the rest of the code and check if it is behaving exactly as expected. Each unit is tested separately before integrating them into modules to test the interfaces between modules. [10]

I have tested each view of the application individually and I made sure that each module is generating the desired output for the corresponding input. There are various modules involved in the application like page navigation, submission of data by the user on to the database, retrieving of data from the database, calculation of the credits for the courses selected by user. I made sure that I have tested each of these modules individually so that an appropriate result is obtained.

Firstly, I made sure that the page navigation is working properly, this involves various buttons such as previous, next, back, view courses, submit etc., Button navigation is one of the main functionality to be tested individually making sure the user gets redirected to the desired page. Secondly, the user selects program option as well as courses. I have tested to make sure that the appropriate information is being displayed based on the program option.

Also, the user submission functionality is tested individually as the user submits his profile information to the database and as this was implemented I made sure that the data from the user goes to the appropriate fields of the database. And lastly, the data retrieval functionality is tested making sure that the user gets the desired information from the database. Each record was tested independently for its functionality. I made sure that a student gets his exact profile when he requests and a faculty member gets the exact POS of the student when he enters the ID. Finally I also made sure that when a student submits his POS, it is done properly without causing any further issues.

### **4.1.1 Tool for Unit testing**

NUnit is the most popular automated unit testing frame work which is used in .Net applications. This is an open source product. This is initially ported from JUnit, which is a framework to test Java applications. It is written entirely in C# and is redesigned to take advantage of many .Net language features. The examples of these features include custom attributes and other reflection related capabilities [13]. In order to conduct unit testing using this tool we have to create test cases for functions to test in our web applications and then compile the project after we are done writing test cases and we have to load the project dll file into NUnit GUI.

### **4.1.2 Test cases and Results**

To perform unit testing I have tested a few of basic functions of my current application. Each of those functions is given below. The five test cases which I have executed against the application using NUnit tool are explained below in detail. I have tested all possible methods using NUnit testing and for the rest of the methods I have done manual testing. I have included all the test cases into one test suite and executed all of them at a time.

#### Test\_Valid\_Login:

This test case is used validate the functional unit in the application in order to verify the user login information, i.e., if the username and password are matching correctly and is a valid login. I have used two test cases one with input as valid login and the other input as invalid login.

#### Test\_Invalid\_Login:

This test case is used to validate the functional unit which makes sure that the user is not allowed to login if there is a mismatch in the username and password supplied by the user. I have used two test cases for this method, one with a valid input and the other with an invalid input.

#### Test\_NoUser:

This test case is used to check if there is no user with that particular user id. I have used two test cases with two valid inputs (i.e., with inputs being those users whose account do not exist in the database) and the tests were passed and one invalid input like already existing user.

#### Test\_Total\_Credits:

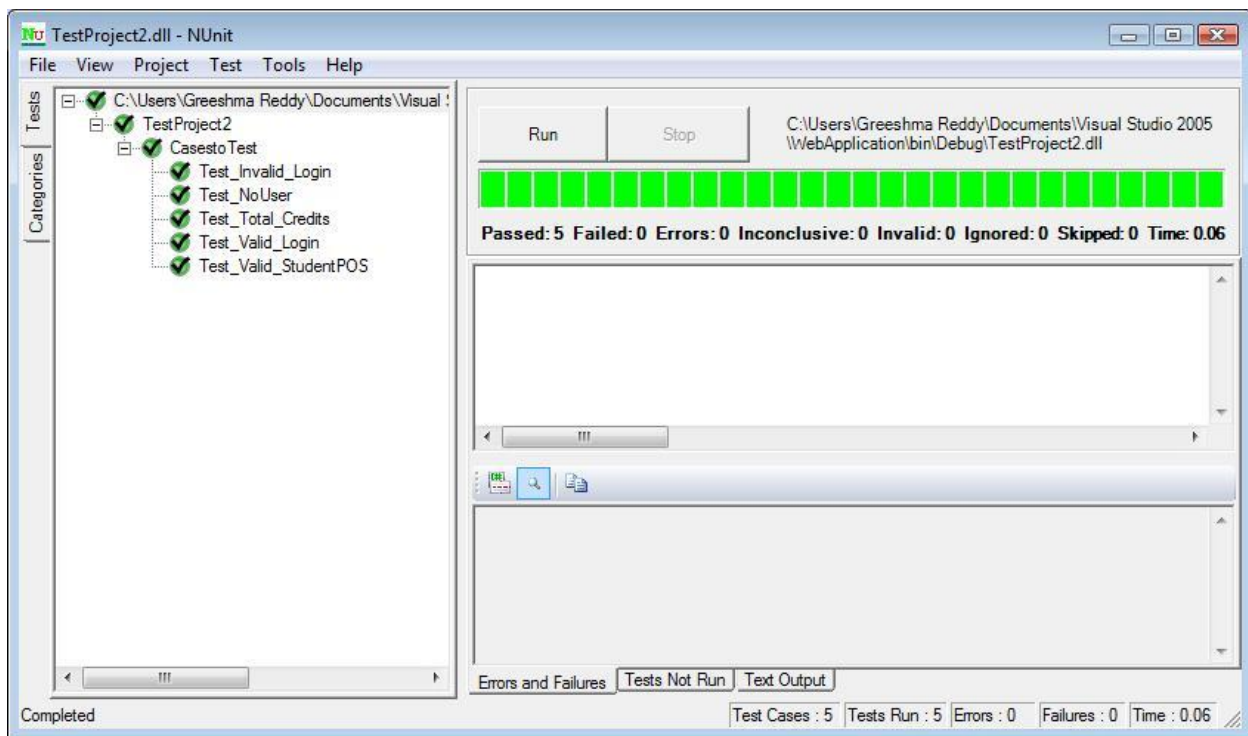
This test case is used to validate the functional unit of the application that checks the total number of credits calculated appropriately for the courses selected by the user. I have used two test cases for the method, one test case with inputs in which user selects all the courses and the other test case in which user selects only core courses.

### Test Valid StudentPOS:

This test case is used to verify if a faculty member gets the exact POS of the student when he enters the student's ID. I have used two test cases for this method one with a valid student ID and the other with an invalid (non-existing) ID.

Apart from these test cases I have also tested various other scenarios manually, like when a user creates an account, I made sure that his/ her details are stored in the database. And in the same way, when he/she selects courses, I verified the database to see if the fields are filled in properly and appropriately.

Figure 4.1 Screen Shot of NUnit Test case Execution



## 4.2 Performance Testing

Software performance testing is used to determine the speed or effectiveness of a software program or device. Qualitative attributes of software such as reliability, scalability and interoperability may also be evaluated during this testing. [10]The performance of a website comes into picture when some hundreds or thousands of users are accessing the website and performance testing ensures that all the users are getting efficient results in less time. By performance testing we can estimate the maximum number of users accessing the web site simultaneously and by means of these testing results one can analyze the measures to further

improve the performance of the application. The tool for performance testing and the testing results are described below.

#### **4.2.1 Tool for performance testing**

To test the performance of the website, I have used Apache JMeter which is an open source software and a 100% pure Java application which is been designed to load test functional behavior and measure performance. This can be used to test performance on static and dynamic resources such as files, databases and queries and more. It can also be used to simulate heavy load on a server, network or to test the strength or to analyze the overall performance under different load types. [11]

#### **4.2.2 System Configuration**

The details of the configuration of the system which I have used in testing the application are listed in the table below:-

Table 4.1 System Configuration

Operating System	Windows Vista
RAM	2GB
Processor	Intel Core 2 Duo
Processor Speed	1.50GHz

#### **4.2.3 Test plan**

All the tests that are listed below were conducted on the system with the above mentioned configuration. The test plan is given below:-

I have taken three test cases into consideration; the three different test cases were conducted differently with different number of users. For test case-1 I have considered 200 users, for test case-2 I have considered 90 users and lastly for test case-3 I have considered like 300 users. Evaluation of the results is based on two factors viz., throughput and response times. There are several factors which made me to put a limit on number of users I was considering and they are described in the future sections below.

The following web pages of my application are tested:-

1. Home(login) Page using HTTP protocol
2. Selection of the courses by student using HTTP protocol

3. Faculty main page, where a faculty member enters the student's WID using HTTP protocol.

## **4.2.4 Test plans Results and Evaluation**

### **4.2.4.1 Test Case 1**

Firstly I have tested the home page to analyze its behavior and to get an idea about the performance of the application. I plan to show the through put as well as response times of the web page when like more number of users are accessing it in order to estimate the performance of the application and how they are affected and thereby estimating the maximum number of users accessing the application at same point of time.

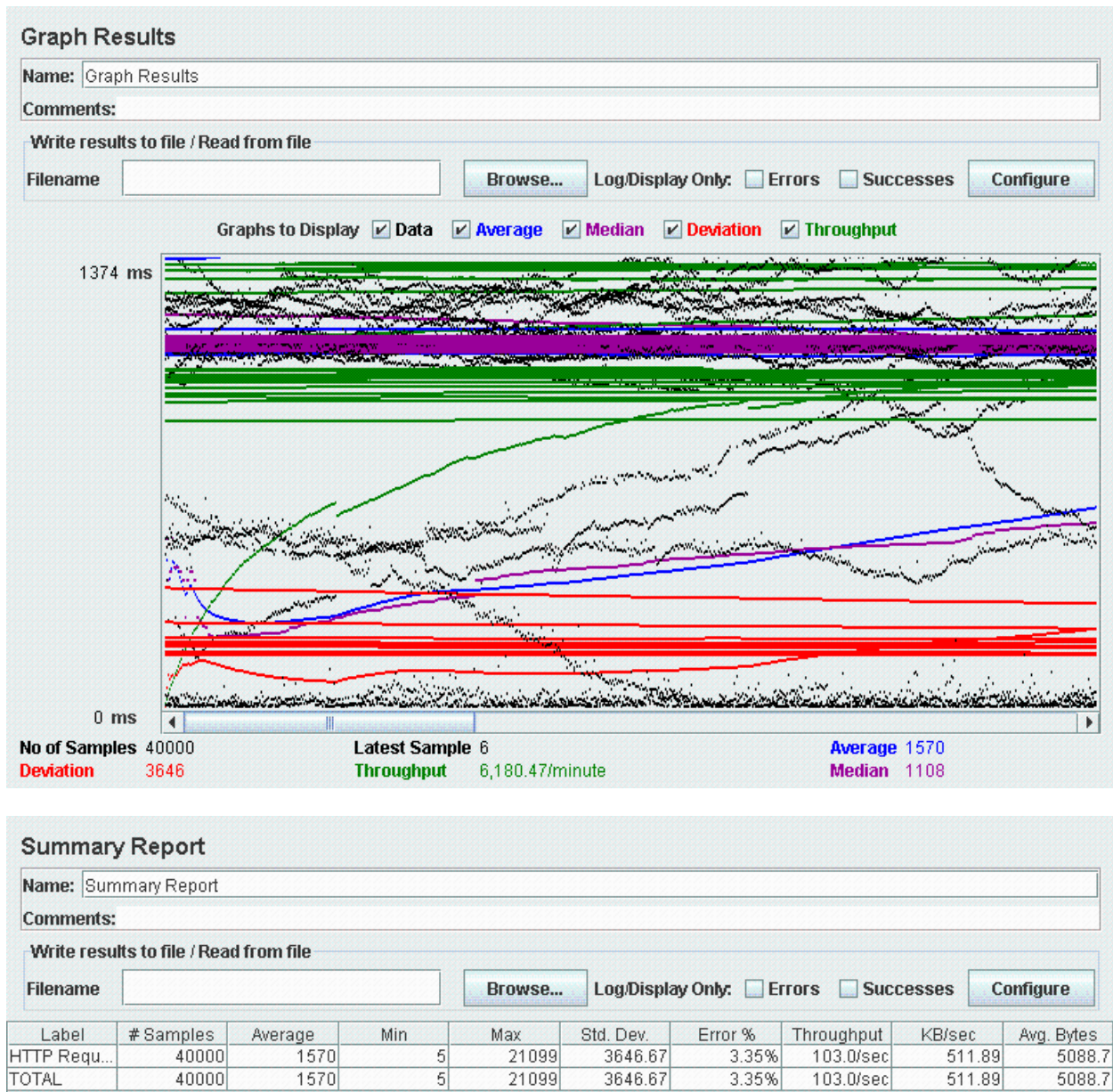
I initially started to conduct the test with 100, 150 and 200 users and with loop count as 100 and observed the results. Then I have noticed that beyond 200 users the JMeter application could not handle. Hence, I have conducted the test with 200 users, loop count as 200 and ramp up period (sec) as 5. It can be noticed from the graph below that the average response time is low (which is 1570ms) and the through put is also high. As this involves communication with database to verify user details, when he passes the username and password this will affect the response as well as the through put time.

I have used the HTTP protocol while calling this web page. I tried to further estimate the performance of the home page by increasing the number of users but it failed to handle more number of users and it often gave Java.lang.OutOfMemory error and the JMeter application would stop and also I have noticed a decrease in performance with increasing the number of users beyond 200 hence I limited the number to 200 for this particular test case. I have shown the results of the graph involving the number of users to be 200, ramp-up period as 5 sec and the loop count as 200.

The graph below explains the above all characteristics of the web page. [12] The graph shown below gives an idea about the test results of the home (login) page.



Figure 4.2 Graph results and Summary report for Home page.



#### 4.2.4.2 Test Case 2

The second test case I have planned to estimate the performance is the courses page where the student chooses the program option as well as core and non-core courses.

I plan to show to show the through put and response time when like many users are accessing the application and how they are affected and also to estimate the maximum number of users accessing the application simultaneously.

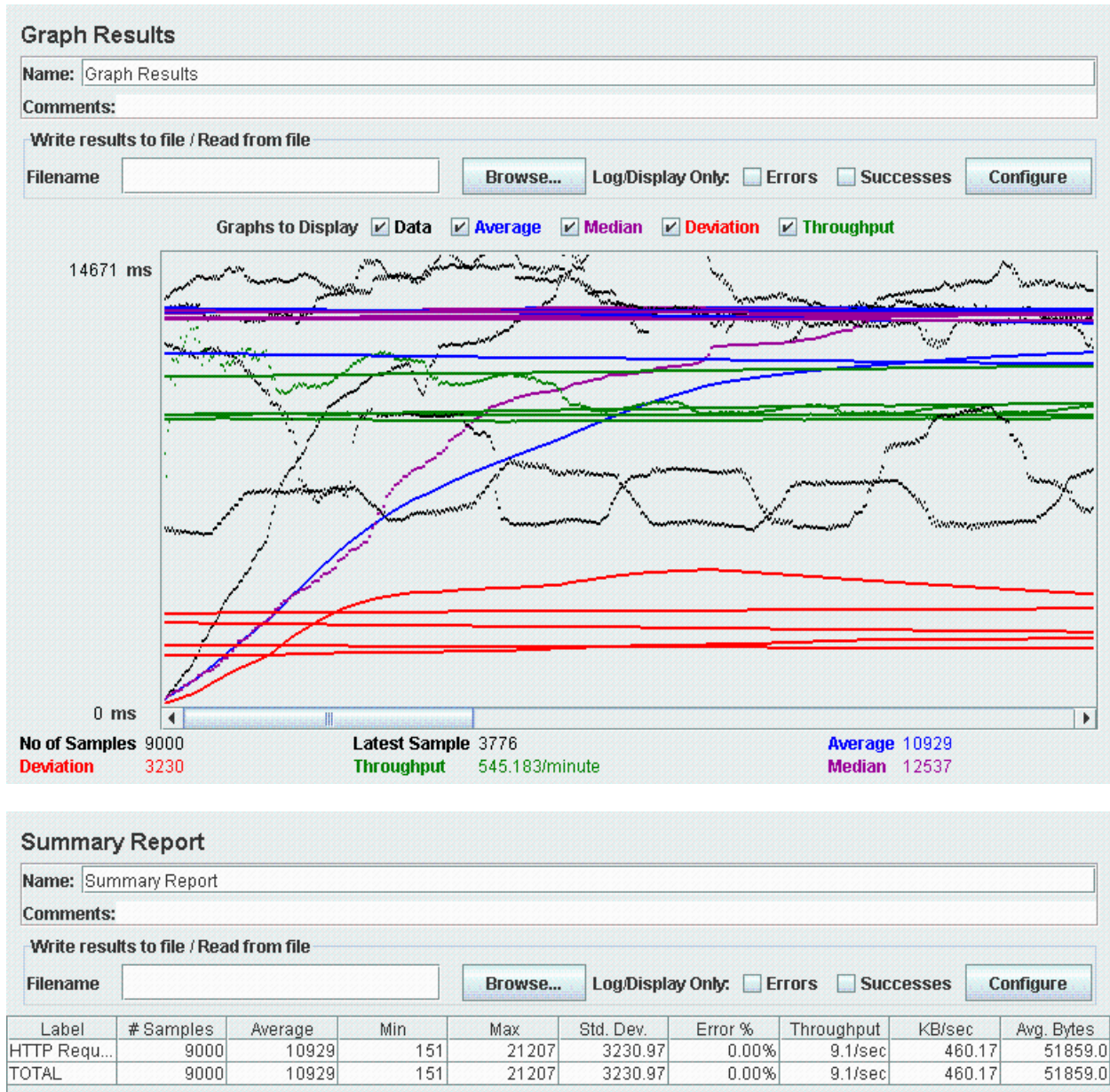
I initially started considering the number of users to be 100 and with the loop count to be 100 with ramp up period being 5 sec. I got the Java.lang.OutOfMemory error and the JMeter application stopped eventually. Hence, I have limited the number of users to 90 and the loop count to 100 with ramp up period (in sec) to 5. From the graph shown below it can be noticed that the through put is considerably moderate and is 545/ minute and also with moderate response time being 10929 ms.

As I have made use of AJAX controls in this web page to display the variation in appearance of the courses that can be selected by a student with varying program option, hence it can be noticed that the through put and response time are affected. This puts an additional overhead hence there is an influence on the through put and response time.

Using Ajax technology has a great influence on the through put and the response times. The same can be observed form the graphs below. I have shown the results of the graph involving the number of users to be 90, ramp-up period as 5 sec and the loop count as 100.

The graph results of the degreeform courses page can be seen below with through put and the average response time and also with other characteristics. The same can be observed in the summary report shown below.

Figure 4.3 Graph results and Summary report for courses selection page.



### 4.2.4.3 Test Case 3

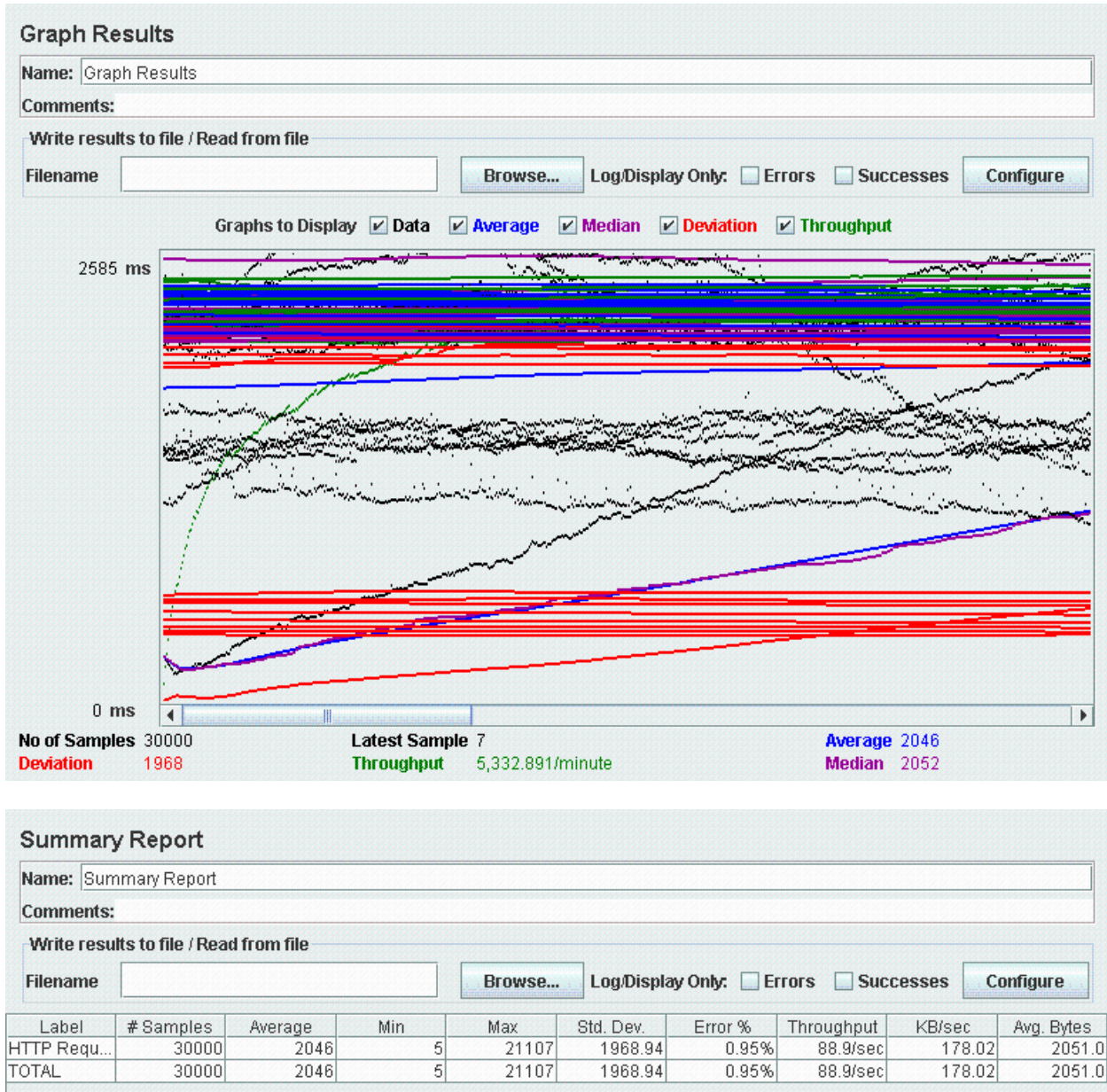
Lastly, the third page I have tested is the faculty page. In this page, a member of faculty enters the student WID in order to view the profile of that particular student having that ID. I plan to show the performance of the application by showing the through put and response time when like many users are trying to access the application and how they are affected and also maximum number of users using the application at same point of time.

I carried out testing considering initially with 100, 200,300 users and loop count to 100 with ramp-up period being 5sec and noticed the variation in the throughput and the response times with increase in number of the users. When I tried to increase the number of users beyond 300, I got the Java.lang.OutOfMemory error and the JMeter application stopped eventually hence I have limited the number of users to 300.

Then I conducted the test using number of users as 300, the loop count to 100 and I have set up the ramp up period to 5 sec. It can be noticed from the graphs shown below that the through put is high (which is 5333/min) and the average response time is low which is 2046 ms. This page involves establishing connection to the database to match the student's wildcat ID and to retrieve the student's profile from the database and hence it affects the through put and response time. This puts an additional overhead to establish the connection to database hence there is an influence on the through put and response time.

I have shown the results of the graph involving the number of users to be 300, ramp-up period as 5 sec and the loop count as 100. The same results about the through put and response time can be obtained from the summary report shown below.

Figure 4.4 Graph results and Summary report for faculty page.



The table shown above gives the summary report for the faculty page showing the throughput, average response time etc.,

#### 4.2.4 Testing Analysis

The main purpose of performance testing using JMeter is to analyze the response time requirements of the web based applications. Response time generally refers to the elapsed time when the user requests something or sends request and the time when he obtains what he wanted.

From the various test cases I have conducted and by observing the results I have noticed that the variation in performance of the application depends on several factors. When the number of users is limited to like a less number i.e., 10, 40, 50 etc., the through put and the response time were like considerably good. Increase in number of users had a great impact on the performance of the website. With more number of users I have noticed a decrease in performance which indicates that there has been more load on the application and this would account to several other factors like the hardware resources CPU, RAM and hard disk. There is a chance for performance degradation because of the low CPU processing and by using better CPU configuration it could handle more number of users. CPU time being the important factor in affecting the through put, I have noticed that by concentrating on improving the system configuration I can improve the through put as well as response time. This would prevent the application from crashing when number of users is more. With better processor speed i.e., like more than what I am having currently, 1.5GHz, one can achieve good performance results for more users.

The other factors that affect the through put and the response time would be inclusion of AJAX technology within them, establishing connection with the database etc, by which there can be a high impact on the throughput and average response times.

## **Chapter 5 - Conclusion and Future Work**

### **5.1 Conclusion**

The online student profile management system is been designed for the students and faculty so that the filing of the program of study can be made easier and faster.

The students can easily pick the program option as well as courses from the drop down menus provided and can submit their POS. They can also view the preview of the selection of the courses. Faculty in turn can also view the POS of the student. Implementing of this application gave me a good learning experience and exposure to the technologies which I have used in this project viz., Visual Studio 2005, C#.NET language and SQL Server 2005. Just choosing the courses from the drop down menu would make users comfortable with the application and easy to use.

### **5.2 Over all Experience**

The current application, online student profile management system which is a .NET web application is been developed using ASP.NET 2.0, SQL Server 2005 in the C# programming language. By implementing this project, I have gained a good learning experience with C#.NET programming language. Earlier I had experience working on SQL server management studio and that helped me a lot while working on this project while designing the database, writing SQL queries etc.,.

After gathering all the requirements for the project next thing I concentrated was learning the C#.net language and other Ajax controls. Before starting working on this application I just had a very basic knowledge of C#.Net. So I had to work a lot on learning it and also I never used Ajax controls before in any of my projects, so I spent like a couple of weeks understanding what Ajax is and how it can be helpful in my project. I referred various online tutorials and books to learn about this and worked on few sample applications to make myself comfortable to use it. After it appeared to me that I was comfortable working with these technologies I started off working on my project. I made use of one of the features of Ajax in displaying the appearance of courses to the student with the change in program option i.e., it displays different page for two different program options viz., Thesis and Non-thesis.

It has been a great pleasure for me to work on this challenging project. This project provided me a good practical knowledge of various things like programming in C#.Net web based application and SQL server. It also provided me a good knowledge about the latest technologies in web

based applications. This learning experience helps me a lot in providing guidance to working in my future projects involving these technologies.

Currently I have used Visual Studio 2005 version while doing this project, I came to know about the newer version of Visual studio which is VS 2008 and its salient features after starting the project. I would have used Visual studio 2008 version instead of 2005. As I was comfortable working with VS 2005 I have continued to work on it and similar is the case with SQL server 2005 version. There are like several good features with SQL server 2008 like data compression and back up compression which are not in 2005 version. Using visual studio 2008, the unit testing would have been made easier than using the N Unit tool, as it comes up with in-built tool that can conduct unit testing of the application.

### **5.3 Future Enhancements to the project**

The following changes can be made to the application in future:-

- Currently I have implemented this application that can be used by only CIS majors in our university, this application may be extended to be used by students of other majors of college of engineering.
- The current system is only confined to graduate students. More functionality can be added so that it can be extended to use by under graduate and PhD students also.
- The application can be extended so that it facilitates even the faculty members to create their own profile and non-faculty members to check if a student has filed his/her POS or not.
- Can add more features and refine the current website in order to make it appear prettier and more interactive.



## REFERENCES

- [1] ASP.NET and Web Development Overview,-  
<http://en.wikipedia.org/wiki/ASP.NET>
  
- [2] DOTNET frame work Architecture,  
<http://technotes.towardsjob.com/dotnet>
  
- [3] Visual Studio 2005,-  
[http://msdn.microsoft.com/en-us/library/fx6bk1f4\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/fx6bk1f4(VS.80).aspx)
  
- [4] Microsoft SQL Server,-  
[http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)
  
- [5] AJAX,-  
[http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
  
- [6] Multi tier Architecture Overview,-  
[http://en.wikipedia.org/wiki/Multitier\\_architecture](http://en.wikipedia.org/wiki/Multitier_architecture)
  
- [7] Three tier Architecture,-  
<http://www.beansoftware.com/ASP.NET-Tutorials/Three-Tier-Architecture.aspx>
  
- [8] Use Case Diagrams Overview,-  
[http://en.wikipedia.org/wiki/Use\\_case\\_diagram](http://en.wikipedia.org/wiki/Use_case_diagram)
  
- [9] Class Diagrams Overview,-  
[http://en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram)
  
- [10] Software Testing Overview,-  
[http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing)
  
- [11] JMeter Overview,-  
<http://jakarta.apache.org/jmeter/>
  
- [12] Performance testing guidance with JMeter  
<http://javaboutique.internet.com/tutorials/JMeter/>
  
- [13] Unit Testing  
<http://www.nunit.org/index.php>