

# **LEAVE MANAGEMENT SYSTEM**

A Project Report Submitted in Partial Fulfillment of the requirements of the  
Award of Department of

COMPUTER SCIENCE

**Submitted**

**By**

**Husain Vakhariya (04BCA6C3)**

*Under the Esteemed Guidance of*

**Mr. Pradeep Vanpariya**

**Mr. Keyur Zala**

Department of Computer Science

**Virani Science College**

Department of Computer Science

**Saurashtra University**

**RAJKOT**

*Virani Science College*

---

<b>Table of contents</b>	<b>page no</b>
1. Abstract	4
2. Organization profile	6
3. Scope and Purpose	7
4. System Requirements and analysis	8
4.1. Problem definition	8
4.2 System overview	8
4.2.1. Existing system	8
4.2.2. Proposed system	8
4.3. System architecture	9
4.4. Definitions, acronyms and abbreviations	10
5. Implementation issues	11
5.1. .Net Frame Work	11
5.2. Asp.Net	16
5.3. Ado.Net	18
5.4. Sql Server 2005	20
5.5. Html	22
5.6. IIS	22
6. Software Requirements specifications	26
6.1. Modular description and functional requirements	27
6.2. Product Perspective	28
6.3. Software Interface	28
6.4. Hardware Interface	29
7. System Design	30
7.1. System Models	32
7.1.1. Usecase Diagrams	33
7.2. Usecase Description and Functionality	36

7.3. Dataflow Diagrams	52
7.4. Class Diagram	60
7.5. Dynamic models	61
7.5.1.Sequence diagrams	61
8. Database Design	69
8.1. Normalization	70
8.2. ER Diagrams	71
8.3. Data Base Tables	73
9. Testing	83
9.1. Types of testing	83
9.2. Test Cases	85
10. Sample screenshots	87
11. Conclusion and Future Scope	104
12. Biblography	105

## **1. ABSTRACT**

### **1.1 OBJECTIVE:**

The main objective of the proposed system is to decrease the paper work and easier record maintenance by having a particular website for attendance, leaves and notice maintenance.

Initially attendance is taken by using the attendance register in the college for staff. Staff needs to submit their leaves manually to their respected higher authorities.

This increases the paper work at the same time maintaining the records is more tedious. Maintaining the notices in the records also increases the paper work

### **MODULES:**

The Proposed system has three main modules

- Staff
- HOD
- Principal

#### **Staff Module**

- Recording attendance

Staff records their attendance at the beginning of each day. Any exceptions (attendance record not present for the previous day) are highlighted when staff logs in on the next day. Staff can enter a justification for the exception and it is forwarded to supervisor for approval.

- View Attendance Report

Staff can query on their attendance for a given period.

- Apply Leave

Staff can send a leave application to the higher authority with a reason for leave and number of days.

### **HOD Module**

This module handles the administrative functions of a department.

- Attendance reports of staff

Can view and generate attendance reports of all staff members for a period of time.

- Daily Absent Report

HOD can view the total number of absentees of a particular department for a given day. They can also enquire on the attendance history of an individual staff in a particular month.

- View leave applications

If any leave applications received from the department staff, HOD is able to accept/reject those applications and can write a remark for that.

### **Principal Module:**

This module is the higher authority for all

- Attendance reports of staff

Can view and generate attendance reports of all staff members for a period of time.

- Daily Absent Report

HOD can view the total number of absentees of a particular department for a given day. They can also enquire on the attendance history of an individual staff in a particular month.

- View leave applications

If any leave applications received from the department staff, HOD is able to accept/reject those applications and can write a remark for that.

- Registration

Principal can register a staff at the time of recruitment and gives username and password generated automatically by the system.

## **2. Organization profile**

**/////Student's work: Past organization profile here//////////**

## **3. SCOPE & PURPOSE**

### **3.1 PURPOSE:**

The purpose of Attendance Management System is to maintain the attendance records of the staff of a college. This system also maintains the leave information and at the same time it also maintains the notices of the college.

### **3.2 SCOPE:**

Once the staff logged in they need to record their attendance in the proposed system the higher authorities like HOD and Principal can view the attendance details of the staff and leave applications generated by the staff.

## **4. SYSTEM REQUIREMENT AND ANALYSIS**

### **4.1 PROBLEM DEFINATION:**

As the existing system increases the paper work and record maintenance tedious there is a need of new system which makes the record maintenance easy that is the common requirement the college attendance management system. So the completion of this project can successfully eliminate the existing problems with the present system.

### **4.2 SYSTEM OVERVIEW**

#### **4.2.1 EXISTING SYSTEM:**

In the current system, an attendance register is maintained at the front-office and employees enter their in and out times. The Hod department does the attendance and leave calculation manually at the end of each month. This increases the paper work and makes the record maintenance tedious.

#### **4.2.2 PROPOSED SYSTEM:**

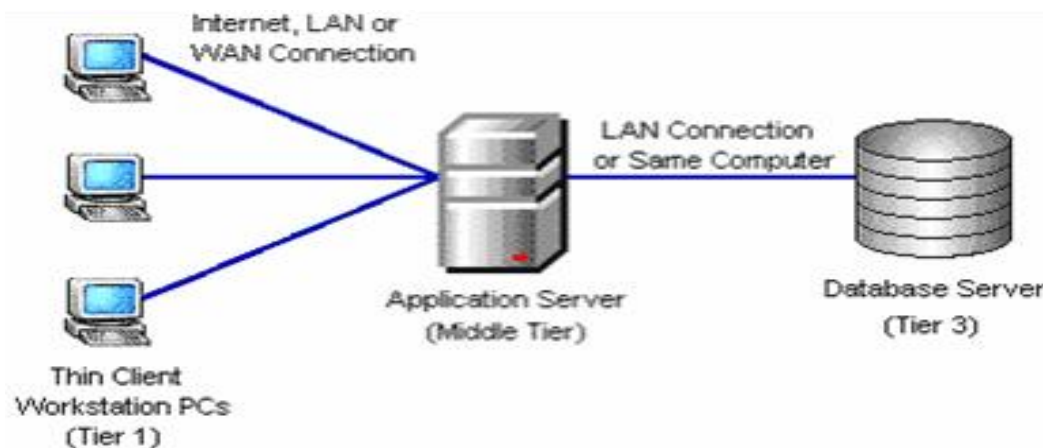
The proposed system automates the existing system. It decreases the paper work and easier record maintenance by having a Database for attendance, leaves and notice maintenance.

The database maintains the information about the attendance, leaves and notice. It reduces the time and manual paper work for apply a leave.

The Leave Management System reduces the over work and risk of the HOD/Principal.

#### 4.3 SYSTEM ARCHITECTURE

The application will follow three-tier architecture. In three-tier architecture application will run the traditional client/server model but from the web server. The client only displays the GUI and data but has no part in producing results.



Three-tier architecture will contain the following tiers

##### **Client/Presentation Tier:**

This tier includes all the HTML content or forms to be displayed on the client browser. It is the form which provides the user interface to end user. Programmer uses this tier to get or set the data back and forth.

##### **Business Logic Layer**

In the Business logic tier, the actual processing of the data and the logic behind the implementation of the application will be present. This tier can contain a class, which can be used to write the functions, and also works as a mediator between the presentation tier and data tiers.

##### **Data Tier:**



Data Tier contains methods and classes that deal with passing and storing data to the data Storage Layer. Queries or stored procedures are used to access the data from the database or to perform any operation to the database. It stores the data passed by the presentation tier.

#### **4.4 DEFINATIONS, ACRONYMS & ABBREVIATIONS**

- HTML: Hypertext Markup Language is a markup language used to design static web pages.
- Asp: Active server pages is used to develop web application
- IIS: Internet Information Service is a web server to run web application
- VS :Visual Studio is application where we can develop application by using this IDE
- HTTP: Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server.
- HTTPS: Secure Hypertext Transfer Protocol is a HTTP over SSL (secure socket layer).
- TCP/IP: Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

#### **REFERENCES**

- IEEE SRS Format

#### **TECHNOLOGIES**

- .Net Framework: Application Architecture
- Sqlserver: Database
- Visual Studio 2008: Development Tool
- IIS: Web Server
- MS Visio: Design Too

## 5. IMPLEMENTATION ISSUES

### 5.1 Microsoft. NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time,

providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

### **Features of the Common Language Runtime:**

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® MS Access™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

### **.NET Framework Class Library**

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is Object Oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.
- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

## Client Application Development

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system (such as GUI elements and limited file access) without being able to access or compromise other resources. Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Your applications can implement the features of a local application while being deployed like a Web page.

## 5.2 ASP.NET

ASP.NET is part of the whole .NET framework, built on top of the Common Language Runtime (also known as the CLR) - a rich and flexible architecture, designed not just to cater for the needs of developers today, but to allow for the long future we have ahead of us. What you might not realize is that, unlike previous updates of ASP, ASP.NET is very much more than just an upgrade of existing technology – it is the gateway to a whole new era of web development.

ASP.NET is a feature at the following web server releases

- Microsoft IIS 5.0 on WINDOWS 2000 Server
- Microsoft IIS 5.1 on WINDOWS XP

ASP.NET has been designed to try and maintain syntax and run-time compatibility with existing ASP pages wherever possible. The motivation behind this is to allow existing ASP Pages to be initially migrated ASP.NET by simply renaming the file to have an extension of .aspx. For the most part this goal has been achieved, although there are typically some basic code changes that have to be made, since VBScript is no longer supported, and the VB language itself has changed.

Some of the key goals of ASP.NET were to

- Remove the dependency on script engines, enabling pages to be type safe and compiled.
- Reduce the amount of code required to develop web applications.
- Make ASP.NET well factored, allowing customers to add in their own custom functionality, and extend/ replace built-in ASP.NET functionality.
- Make ASP.NET a logical evolution of ASP, where existing ASP investment and therefore code can be reused with little, if any, change.
- Realize that bugs are a fact of life, as ASP.NET should be as fault tolerant as possible.

### Benefits of ASP.NET

The .NET Framework includes a new data access technology named ADO.NET, an evolutionary improvement to ADO. Though the new data access technology is evolutionary, the classes that make up ADO.NET bear little resemblance to the ADO objects with which you might be familiar. Some fairly significant changes must be made to existing ADO applications to convert them

to ADO.NET. The changes don't have to be made immediately to existing ADO applications to run under ASP.NET, however.

ADO will function under ASP.NET. However, the work necessary to convert ADO applications to ADO.NET is worthwhile. For disconnected applications, ADO.NET should offer performance advantages over ADO disconnected record sets. ADO requires that transmitting and receiving components be COM objects. ADO.NET transmits data in a standard XML-format file so that COM marshaling or data type conversions are not required.

#### **ASP.NET has several advantages over ASP.**

The following are some of the benefits of ASP.NET:

- Make code cleaner.
- Improve deployment, scalability, and reliability.
- Provide better support for different browsers and devices.
- Enable a new breed of web applications.

#### **ActiveX**

ActiveX is a specification developed by Microsoft that allows ordinary Windows programs to be run within a Web page. ActiveX programs can be written in languages such as Visual Basic and they are compiled before being placed on the Web server.

ActiveX application, called controls, are downloaded and executed by the Web browser, like Java applets. Unlike Java applets, controls can be installed permanently when they are downloaded; eliminating the need to download them again. ActiveX's main advantage is that it can do just about anything.

This can also be a disadvantage:

Several enterprising programmers have already used ActiveX to bring exciting new capabilities to Web page, such as "the Web page that turns off your computer" and "the Web page that formats disk drive".

Fortunately, ActiveX includes a signature feature that identifies the source of the control and prevents controls from being modified. While this won't prevent a control from damaging system, we can specify which sources of controls we trust.



ActiveX has two main disadvantages

It isn't as easy to program as scripting language or Java.

ActiveX is proprietary.

It works only in Microsoft Internet Explorer and only Windows platforms.

### **5.3 ADO.NET**

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed via OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

ADO.NET cleanly factors data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET Dataset object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remote between tiers. The ADO.NET Dataset object can also be used independently of a .NET data provider to manage data local to the application or sourced from XML.

#### **Why ADO.NET?**

As application development has evolved, new applications have become loosely coupled based on the Web application model. More and more of today's applications use XML to encode data to be passed over network connections. Web applications use HTTP as the fabric for communication between tiers, and therefore must explicitly handle maintaining state between requests. This new model is very different from the connected, tightly coupled style of programming that characterized the client/server era, where a connection was held open for the duration of the program's lifetime and no special handling of state was required.

In designing tools and technologies to meet the needs of today's developer, Microsoft recognized that an entirely new programming model for data access was needed, one that is built upon the .NET Framework. Building on the .NET Framework ensured that the data access technology

would be uniform—components would share a common type system, design patterns, and naming conventions.

ADO.NET was designed to meet the needs of this new programming model: disconnected data architecture, tight integration with XML, common data representation with the ability to combine data from multiple and varied data sources, and optimized facilities for interacting with a database, all native to the .NET Framework.

### **Leverage Current ADO Knowledge**

Microsoft's design for ADO.NET addresses many of the requirements of today's application development model. At the same time, the programming model stays as similar as possible to ADO, so current ADO developers do not have to start from scratch in learning a brand new data access technology. ADO.NET is an intrinsic part of the .NET Framework without seeming completely foreign to the ADO programmer.

ADO.NET coexists with ADO. While most new .NET applications will be written using ADO.NET, ADO remains available to the .NET programmer through .NET COM interoperability services.

ADO.NET provides first-class support for the disconnected, n-tier programming environment for which many new applications are written. The concept of working with a disconnected set of data has become a focal point in the programming model. The ADO.NET solution for n-tier programming is the Dataset.

### **XML Support**

XML and data access are intimately tied—XML is all about encoding data, and data access is increasingly becoming all about XML. The .NET Framework does not just support Web standards—it is built entirely on top of them.

## **5.4 SQL SERVER 2005**

Microsoft SQL Server 2005 is comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. SQL Server 2005 allows companies to gain

greater insight from their business information and achieve faster results for a competitive advantage.

## **Top-10 Features of SqlServer-2005**

### **1. T-SQL (Transaction SQL) enhancements**

T-SQL is the native set-based RDBMS programming language offering high-performance data access. It now incorporates many new features including error handling via the TRY and CATCH paradigm, Common Table Expressions (CTE), which return a record set in a statement, and the ability to shift columns to rows and vice versa with the PIVOT and UNPIVOT commands.

### **2. CLR (Common Language Runtime)**

The next major enhancement in SQL Server 2005 is the integration of a .NET compliant language such as C#, ASP.NET or VB.NET to build objects (stored procedures, triggers, functions, etc.). This enables you to execute .NET code in the DBMS to take advantage of the .NET functionality. It is expected to replace extended stored procedures in the SQL Server 2000 environment as well as expand the traditional relational engine capabilities.

### **3. Service Broker**

The Service Broker handles messaging between a sender and receiver in a loosely coupled manner. A message is sent, processed and responded to, completing the transaction. This greatly expands the capabilities of data-driven applications to meet workflow or custom business needs.

### **4. Data encryption**

SQL Server 2000 had no documented or publicly supported functions to encrypt data in a table natively. Organizations had to rely on third-party products to address this need. SQL Server 2005 has native capabilities to support encryption of data stored in user-defined databases.

### **5. SMTP mail**

Sending mail directly from SQL Server 2000 is possible, but challenging. With SQL Server 2005, Microsoft incorporates SMTP mail to improve the native mail capabilities. Say "see-ya" to Outlook on SQL Server!

### **6. HTTP endpoints**

You can easily create HTTP endpoints via a simple T-SQL statement exposing an object that can be accessed over the Internet. This allows a simple object to be called across the Internet for the needed data.

#### **7. Multiple Active Result Sets (MARS)**

MARS allow a persistent database connection from a single client to have more than one active request per connection. This should be a major performance improvement, allowing developers to give users new capabilities when working with SQL Server. For example, it allows multiple searches, or a search and data entry. The bottom line is that one client connection can have multiple active processes simultaneously.

#### **8. Dedicated administrator connection**

If all else fails, stop the SQL Server service or push the power button. That mentality is finished with the dedicated administrator connection. This functionality will allow a DBA to make a single diagnostic connection to SQL Server even if the server is having an issue.

#### **9. SQL Server Integration Services (SSIS)**

SSIS has replaced DTS (Data Transformation Services) as the primary ETL (Extraction, Transformation and Loading) tool and ships with SQL Server free of charge. This tool, completely rewritten since SQL Server 2000, now has a great deal of flexibility to address complex data movement.

#### **10. Database mirroring**

It's not expected to be released with SQL Server 2005 at the RTM in November, but I think this feature has great potential. Database mirroring is an extension of the native high-availability capabilities. So, stay tuned for more details....

#### **INFORMATION SUPER HIGHWAY:**

A set of computer networks, made up of a large number of smaller networks, using different networking protocols. The world's largest computing network consisting of over two million computers supporting over 20 millions users in almost 200 different countries. The Internet is growing a phenomenal rate between 10 and 15 percent. So any size estimates are quickly out of date.

Internet was originally established to meet the research needs of the U.S Defence Industry. But it has grown into a huge global network serving universities, academic researches, commercial interest and Government agencies, both in the U.S and Overseas. The Internet uses TCP/IP protocols and many of the Internet hosts run the Unix Operating System.

## **5.5 HTML**

HTML (Hyper Text Mark up Language) is the language that is used to prepare documents for online publications. HTML documents are also called Web documents, and each HTML document is known as Web page.

A page is what is seen in the browser at any time. Each Web site, whether on the Internet or Intranet, is composed of multiple pages. And it is possible to switch among them by following hyperlinks. The collection of HTML pages makes up the World Wide Web.

A web pages is basically a text file that contains the text to be displayed and references of elements such as images, sounds and of course hyperlinks to other documents. HTML pages can be created using simple text editor such as Notepad or a WYSIWYG application such as Microsoft FrontPage.

In either case the result is a plain text file that computers can easily exchange. The browser displays this text file on the client computer.

"Hypertext" is the jumping frog portion. A hyperlink can jump to any place within your own page(s) or literally to anyplace in the world with a 'net address (URL, or Uniform Resource Locator.) It's a small part of the html language.

## **5.6 INTERNET INFORMATION SERVER (IIS):**

A web server is a program connected to the world wide web(www) that furnishes resources from the web browser.

Microsoft IIS is a web server integrated with Windows.NET server that makes it easy to publish information and bring business application to the web.

Because of its tight integration with windows NT server, IIS guarantees the network administrator and application developer the same security, Networking and administrator functionality as windows NT server. Above and beyond its use of familiar Windows NT server

Tools and functionality, IIS also has built-in capabilities to help administer secure websites, and to develop server-intensive web application.

#### **FEATURES OF IIS:**

IIS provides integrated security and access to a wide range of content, work seamlessly with COM components, and has a graphical interface-the Microsoft Management Console (MMC) –that you can use to create and manage your ASP application.

#### **IIS Provides Integrated Security:**

On the internet, most sites allow anybody to connect to the site. The exceptions are commercialists where you pay a one time, monthly fee to access the site. Sites that are restrict the access called secured site. Secured site use either integrated security or login, password security. IIS support both of these methods.

#### **IIS provides Access to Content:**

All web servers can deliver HTML files, but they differ widely in how they treat other types of content. Most servers let you add and modify Multi-purpose Internet Mail Extensions (MMIE) types, but integrate directly into the windows registry. That means IIS natively understands how to treat most common windows file format, such as text (TXT) files, application initialization (INI) files, executable (EXE) files and many others

#### **IIS provides an Interface FOR COM**

You can control many parts of IIS using COM>IIS exposes many of the server's configuration settings via the IIS Admin objects. These objects are accessible from ASP and other languages. That means you can adjust server configuration and create virtual directories and webs programmatically. IIS 4 and higher store settings and web information in a spoil database called the Metaphase. You can use the IIS Admin objects to create new sites and virtual directories be alter the properties of existing sites and virtual directories.

#### **IIS ARCHITECTURES OVERVIEW:**

IIS is a core product, which means that it is designed to work closely with many other products, including all products in the Windows NT Server 4.0 Option pack. The following figure shows the relationship between IIS and other products installed as part of the Windows NT Server 4.0 Option pack.

## **SECURITY FOR IIS APPLICATION**

IIS provides three authentication schemes to control access to ITS resources: Anonymous, Basic and Windows NT challenge/Response. Each of these schemes had different effect on the security context of an application launched by ITS. This includes ISAPI extension agents, COT applications, IDC scripts and future scripting capabilities.

## **ACCESS PRIVIEGES**

IIS provides several new access levels. The following values can set the type of access allowed to specific directories:

- Read
- Write
- Script
- Execute
- Log Access
- Directory Browsing.

## **IIS WEBSITE ADMINISTRATION**

Administering websites can be time consuming and costly, especially for people who manage large internet Service Provider (ISP) Installations. To save time and money Sip's support only large company web siesta the expense of personal websites. But is there a cost-effective way to support both? The answer is yes; if you can automate administrative tasks and let users administer their own sites from remote computers. This solution reduces the amount of time and money it takes to manually administer a large installation, without reducing the number of web sites supported.

Microsoft Internet Information server (IIS) version 4.0 offers technologies to do this:

1. Windows scripting Host (WSH)
2. IIS Admin objects built on top of Active Directory service Interface(ADS))

With these technologies working together behind the scenes, the person can administers sites from the command line of central computer and can group frequently used commands in batch

files. Then all user need to do is run batch files to add new accounts, change permissions, add a virtual server to a site and many other tasks.

## **6. SOFTWARE REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a complete description of the behavior of the software to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. In addition to use cases, the SRS contains functional requirements, which define the internal workings of the software: that is, the calculations, technical details, data manipulation and processing, and other specific functionality that shows how the use cases are to be satisfied. It also contains nonfunctional requirements, which impose constraints on the design or implementation (such as performance requirements, quality standards or design constraints).

The SRS phase consists of two basic activities:

### **1) Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understanding the problem, the goal and constraints.

### **2) Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

### **Role of SRS:**

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.



## **6.1 MODULAR DESCRIPTION & FUNCTIONAL REQUIREMENTS**

### **MODULES:**

The Proposed system has three main modules

- Staff
- HOD
- Principal

#### **Staff Module**

- Recording attendance

Staff records their attendance at the beginning of each day. Any exceptions (attendance record not present for the previous day) are highlighted when staff logs in on the next day. Staff can enter a justification for the exception and it is forwarded to supervisor for approval.

- View Attendance Report

Staff can query on their attendance for a given period.

- Apply Leave

Staff can send a leave application to the higher authority with a reason for leave and number of days.

#### **HOD Module**

This module handles the administrative functions of a department.

- Attendance reports of staff

Can view and generate attendance reports of all staff members for a period of time.

- Daily Absent Report

HOD can view the total number of absentees of a particular department for a given day. They can also enquire on the attendance history of an individual staff in a particular month.

- View leave applications

If any leave applications received from the department staff, HOD is able to accept/reject those applications and can write a remark for that.

### **Principal Module:**

This module is the higher authority for all

- Attendance reports of staff

Can view and generate attendance reports of all staff members for a period of time.

- Daily Absent Report

HOD can view the total number of absentees of a particular department for a given day. They can also enquire on the attendance history of an individual staff in a particular month.

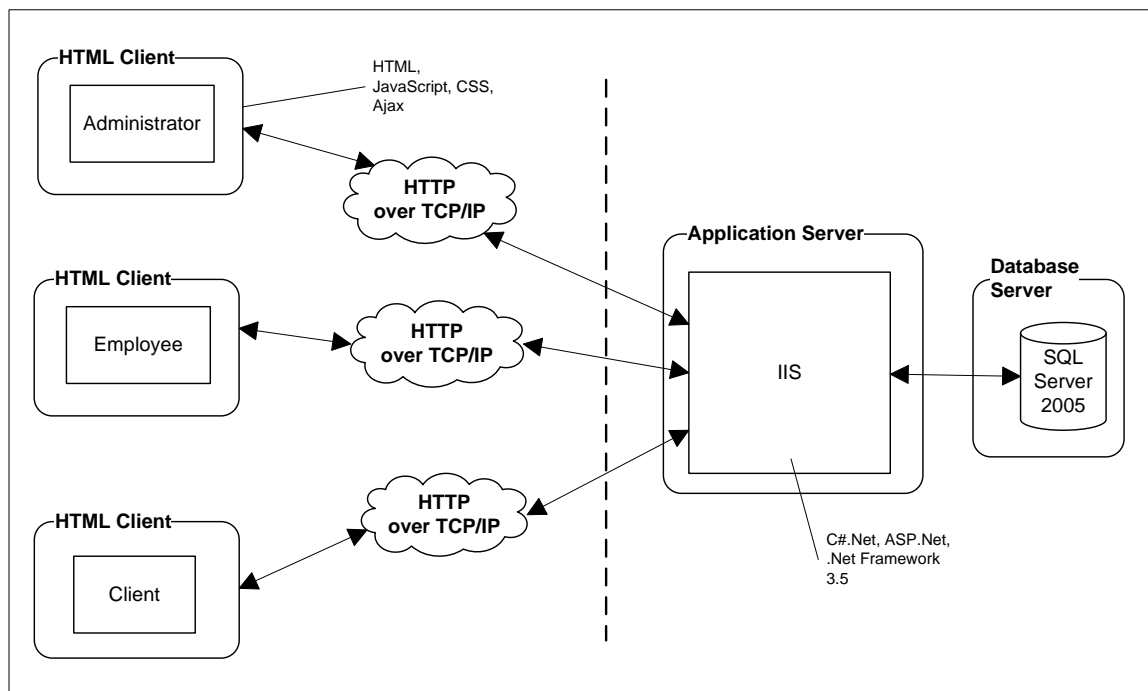
- View leave applications

If any leave applications received from the department staff, HOD is able to accept/reject those applications and can write a remark for that.

- Registration

Principal can register a staff at the time of recruitment and gives username and password generated automatically by the system.

## 6.2 Product Perspective:



- The web pages (ASP) are present to provide the user interface on customer side. Communication between customer and server is provided through HTTP/HTTPS protocols.
- The Client Software is to provide the user interface on system user client side and for this TCP/IP protocols are used.
- On the Server Side web server is for EJB and database server is for storing the information.

## 6.3 Software Interface:

Developer Side:	
Operating System	Windows XP
Application System	IIS 5.4
Database	SQL Server 2005
Programming language	.Net

Development IDE	Visual Studio 2008
Client Side:	
Operating System	Any
Web Browser	Any
Server side:	
Operating system	Any
Application Server	IIS 5.1
DBMS	SQL Server 2005

#### 6.4 Hardware Interface:

Developer side:			
	Processor	RAM	Disk Space
	Intel p4 or equivalent	512MB	2 GB
Client Side:			
	Intel p4 or equivalent	512MB	1 GB
Server Side:			
	Server Environment Capable Hardware	2 GB	As per the Size of the required Data base

#### Communication Interface:

- Client on internet will be using HTTP/HTTPS protocol
- Client on internet will be using TCP/IP protocols

#### Constraints:

- GUI is only in English.

- Login and Password is used for identification of Admin,Employee,Client and there is no facility for guest.
- This System is working for single server.
- There is no Maintainability of backup so availability will get effected.
- Limited to HTTP/HTTPS

## 7. SYSTEM DESIGN

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

### **Unified Modeling Language:**

UML stands for Unified Modeling Language. It is a third generation method for specifying, visualizing and documenting the artifacts of an object oriented system under development. Object modeling is the process by which the logical objects in the real world (problem space) are represented (mapped) by the actual objects in the program (logical or a mini world). This visual representation of the objects, their relationships and their structures is for the ease of understanding. This is a step while developing any product after analysis.

The goal from this is to produce a model of the entities involved in the project which later need to be built. The representations of the entities that are to be used in the product being developed need to be designed.

Software design is a process that gradually changes as various new, better and more complete methods with a broader understanding of the whole problem in general come into existence.

The Unified Modeling Language encompasses a number of models.

- Use case diagrams
- Class diagrams
- Sequence diagrams

### **Use Case Diagram:**

Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from user's perspective.
- The use cases are the functions that are to be performed in the module.
- An actor could be the end-user of the system or an external system.

#### **Class Diagram:**

Class Diagram consists of the classes and the objects and the interaction between them. It mainly deals with the interaction between classes in the system, their behavior and properties of the system. Apart from classes this also provides inheritance relationships in the project. Class diagrams consist of basically two parts: first one is the member variables and class variables and the second part consists of the total number of methods available in the class.

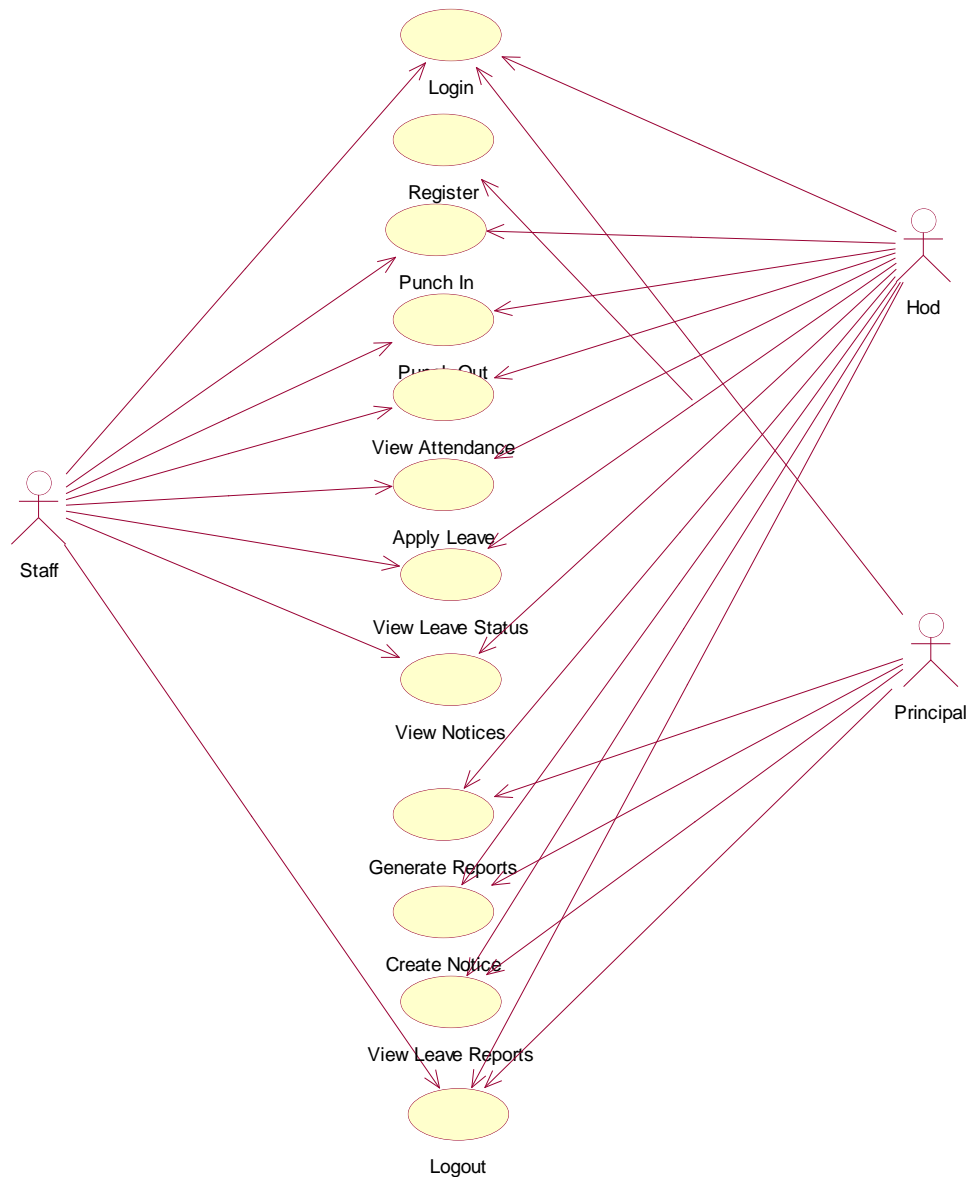
#### **Sequence Diagram:**

The purpose of sequence diagram is to show the flow of functionality through a use case. In other words, we call it a mapping process in terms of data transfers from the actor through the corresponding objects.

The key points are:

- The main purpose is to represent the logical flow of data with respect to a process
- A sequence diagram displays the objects and not the classes.

### **7.1 Use Case Diagrams**



## 7.2 DATA FLOW DIAGRAMS:

, A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and

movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called a "*context diagram*".

### **Context Diagram:**

It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

A DFD is also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

### **DFD SYMBOLS:**

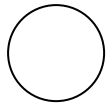
In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows. Data move in a specific direction from an origin to a destination.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

**Symbols**

**Elementary references**

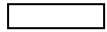




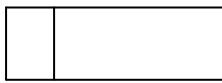
Process that transforms data flow



Data Flow



Source or Destination of data



Data Store

### Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy interface. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

### **Salient features of DFD'S:**

The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.

1. The DFD does not indicate the time factor involved in any process whether the data flow take place daily, weekly or monthly.
2. The sequence of events is not brought out on the DFD.

### **Types of data flow diagrams**

DFD's are of two types

- (a) Physical DFD
- (b) Logical DFD

#### **1. Physical DFD:**

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, and people etc., involved in the current system

#### **2. Logical DFD:**

Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

### **Rules Governing the DFD'S:**

#### **Process**

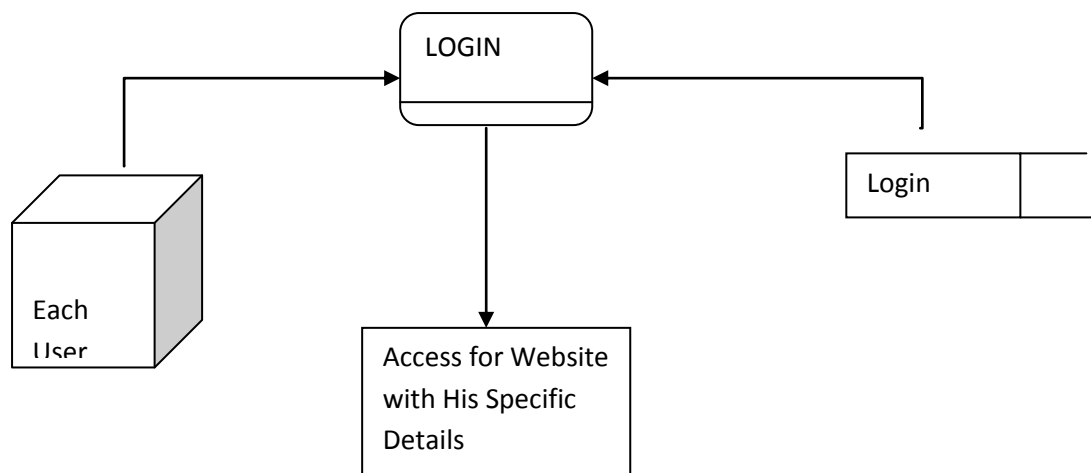
1. No process can have only outputs.

2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase level.

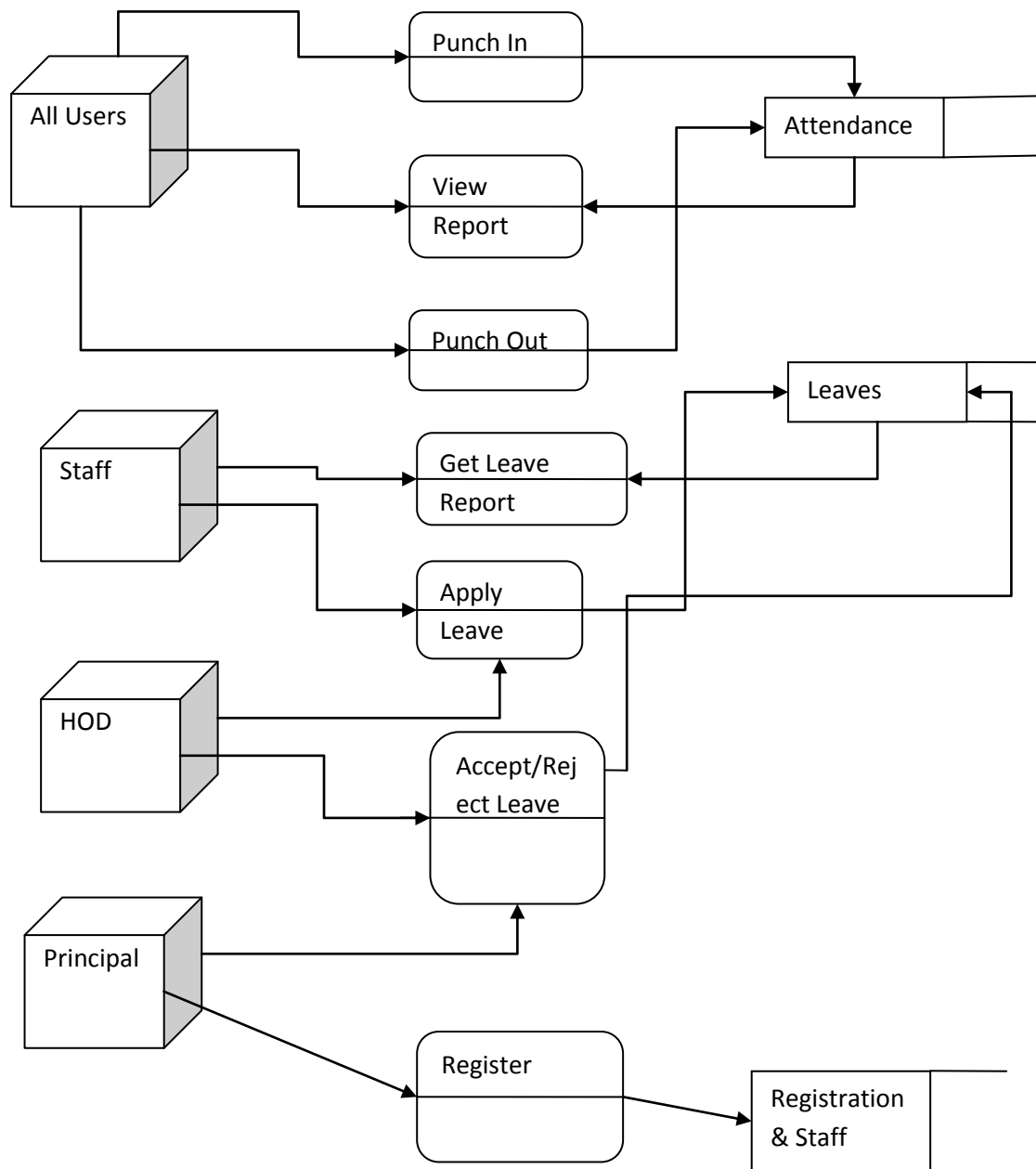
### Data Store

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store, a process, which retrieves, must move data from the source and place the data into data store.
3. A data store has a noun phrase level.

### Level 0:



### Level1:



## Activity Diagram:

Activity diagrams provide a way to model the workflow of a business process. You can also use activity diagrams to model code-specific information such as a class operation. Activity diagrams are very similar to a flowchart because you can model a workflow from activity to activity.

## Activity Diagram Tools:

You can use the following tools on the activity diagram toolbox to model activity diagrams:

- Activities
- Decisions
- End state
- Object
- Object Flow
- Start states
- States
- Swim lanes
- Synchronizations
- Transmissions

### ACTIVITY DIAGRAM FOR STAFF:

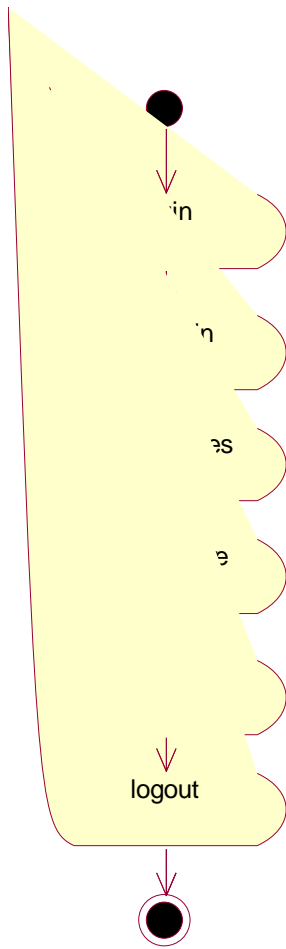


Fig 3.3.3 Activity diagram for staff

## ACTIVITY DIAGRAM FOR HOD:

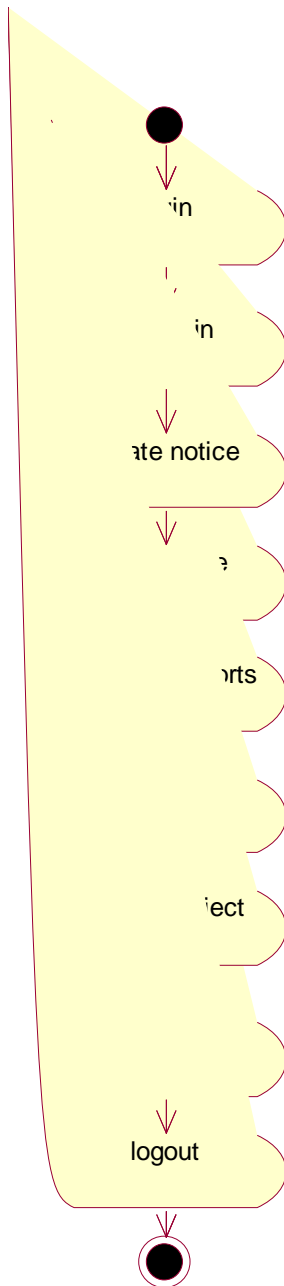


Fig 3.3.4 Activity diagram for Ho

### ACTIVITY DIAGRAM FOR Principal:

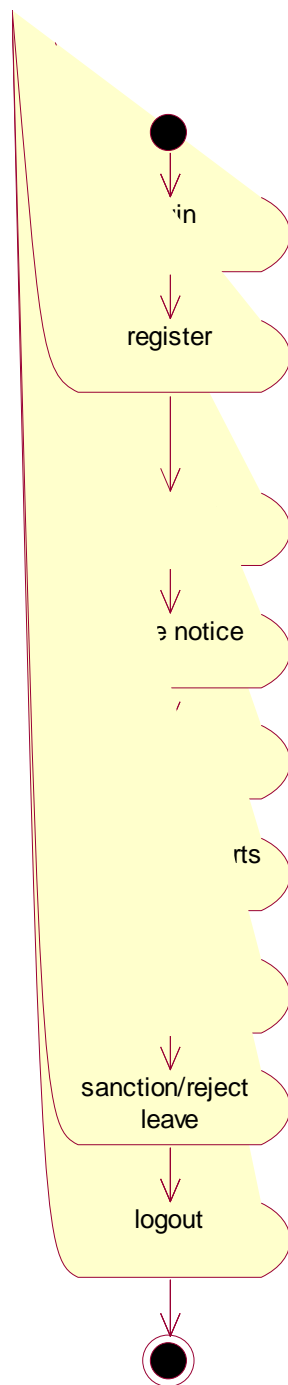


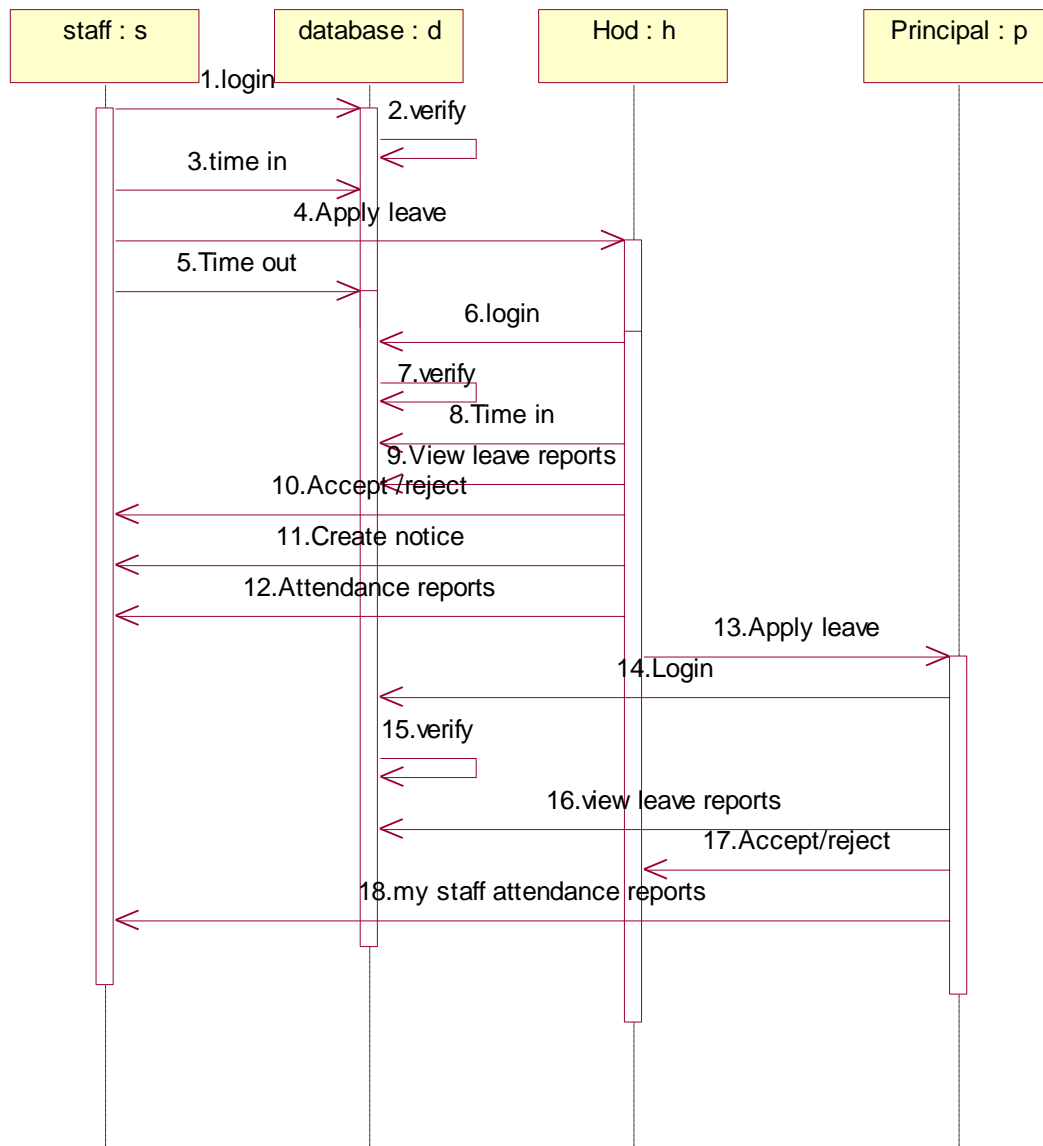
Fig 3.3.5 Activity diagram for Principal

### 7.4 CLASS DIAGRAM



## 7.5 DYANAMIC MODEL

### Sequence Diagrams:



## 8. DATABASE DESIGN

### 8.1 NORMALIZATION

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files. The tables are organized to:

- ② Reduced duplication of data.
- ② Simplify functions like adding, deleting, modifying data etc.,
- ② Retrieving data
- ② Clarity and ease of use
- ② More information at low cost

### Normalization

Normalization is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints on the kind of functional dependencies that could be associated with the relation. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

#### First Normal Form:

A relation R is in first normal form if and only if all underlying domains contained atomic values only.

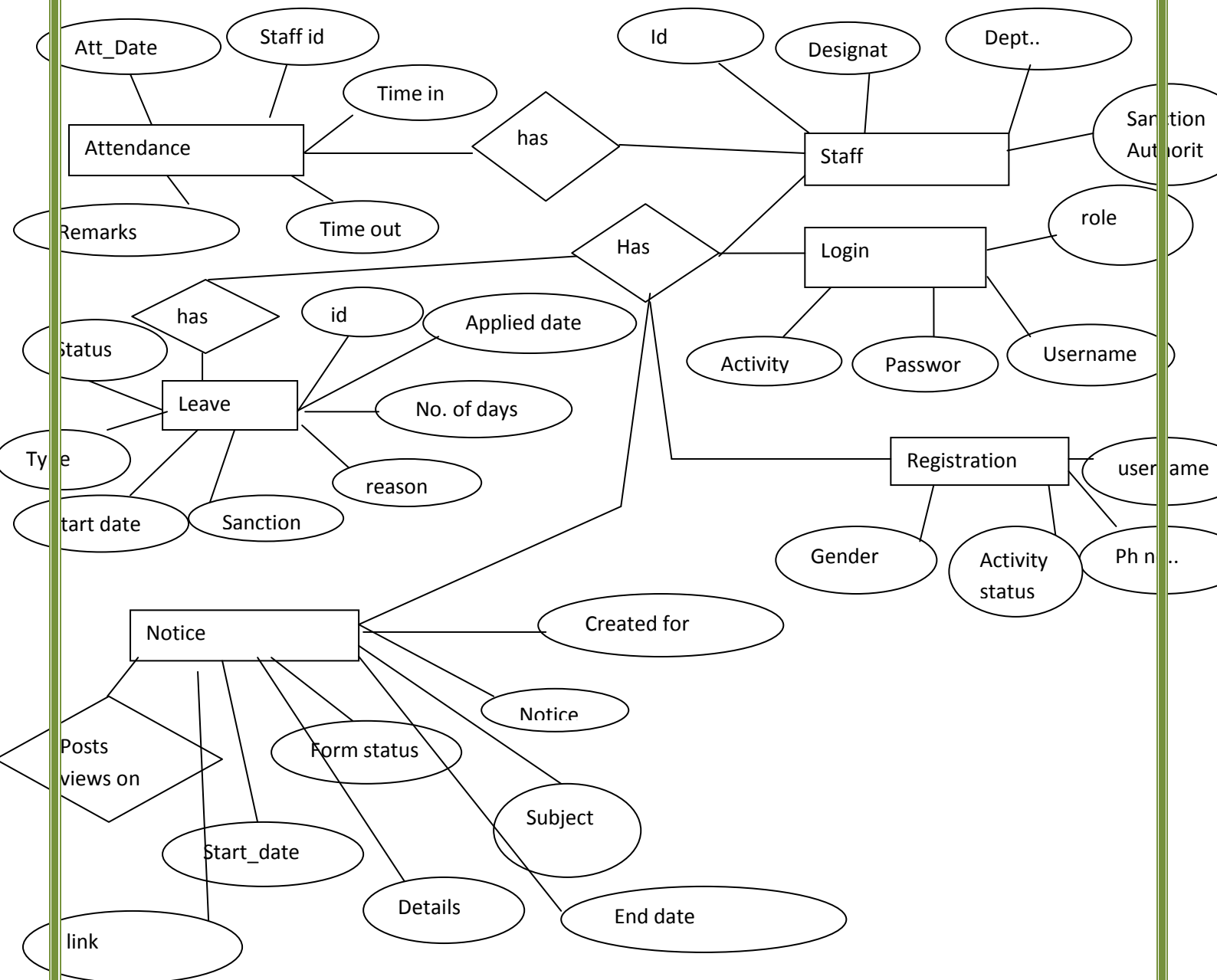
#### Second Normal Form:

A relation R is said to be in second normal form if and only if it is in first normal form and every non-key attribute is fully dependent on the primary key.

#### Third Normal Form:

A relation R is said to be in third normal form if and only if it is in second normal form and every non key attribute is non transitively depend on the primary key.

## 8.2 ER DIAGRAMS:



### 8.3 DATABASE TABLES:

#### Attendance Table:

Column name	Data type	Description
Attendance_date	datetime	Date on which attendance is recorded
Staff_id	Nvarchar(50)	Id of a staff
Attendance_timein	Nvarchar(10)	In time attendance of staff
Attendance_timeout	Nvarchar(10)	Out time attendance of staff
Attendance_Remarks	Nvarchar(200)	Remarks of the staff
Att_timeoutremarks	Nvarchar(200)	Time out remarks

#### Login Table:

Primary key: User Name

Column name	Data type	Description
Username	nvarchar(50)	Username of the staff
Password	nvarchar(20)	Password of the staff
User Role	nvarchar(20)	Role of the user (Lecturer,Hod,principal)

Activity Status	nvarchar(50)	i.e. active or deactivate
-----------------	--------------	---------------------------

### Registration Table:

Column name	Data type	Description
First name	Nvarchar(20)	First Name of staff
Last name	Nvarchar(20)	Last Name of staff
Gender	Nvarchar(10)	Gender of the staff
Address	Nvarchar(200)	Address of the staff
Contact No	bigint	Contact number of staff
DOB	datetime	Date of birth of the staff
email	Nvarchar(30)	Email of the staff
username	Nvarchar(50)	Username of the staff which is automatically generated  (firstname.lastname)
ActivityStatus	nvarchar(50)	Active or deactivate

### Staff Table:

Column name	Data type	Description
Staff_id	nvarchar(50)	Id of the staff

Staff_designation	nvarchar(50)	Designation i.e. (Lecturer , Hod , Principal)
Staff_dept	nvarchar(50)	Department of the staff
Staff_sanctionauthority	nvarchar(50)	Authority for particular staff (i.e. for Hod authority is Principal)

### Leaves Table:

Column name	Data type	Description
Staff_id	Nvarchar(50)	Id of the staff
Leave_type	Nvarchar(10)	Type of leave applying
Leave_applydate	datetime	Leave apply date
Leave_startdate	datetime	Start date of the leave
Leave_days	Int	No of days the leave applied.
Leave_reason	Nvarchar(200)	Reason for the leave
Leave_sanctionauthority	Nvarchar(50)	Leave sanction authority
Leave_remarks	Nvarchar(150)	Remarks for the leave
Leave_enddate	Datetime	End date of the leave
Leave_ClassAdjust	Nvarchar(500)	Adjustment of classes while on the leave.
Leave_Status	Nvarchar(50)	Status of the leave
Leave_Id	Nvarchar(20)	Id of the leave

### Notices Table:

Column name	Data type	Description
Notice_Id	Nvarchar(20)	Id of the notice
Notice_Sub	Nvarchar(50)	Notice subject
Notice_Details	Nvarchar(200)	Details of the notice
Notice_Startdate	Datetime	Start date of the notice
Notice_Enddate	Datetime	End date of the notice
Notice_Link	Nvarchar(50)	Link of the notice
Notice_Createdby	Nvarchar(50)	Creator of the notice
Notice_createdfor	Nvarchar(20)	Receiver of the notice

### Master Leaves Table:

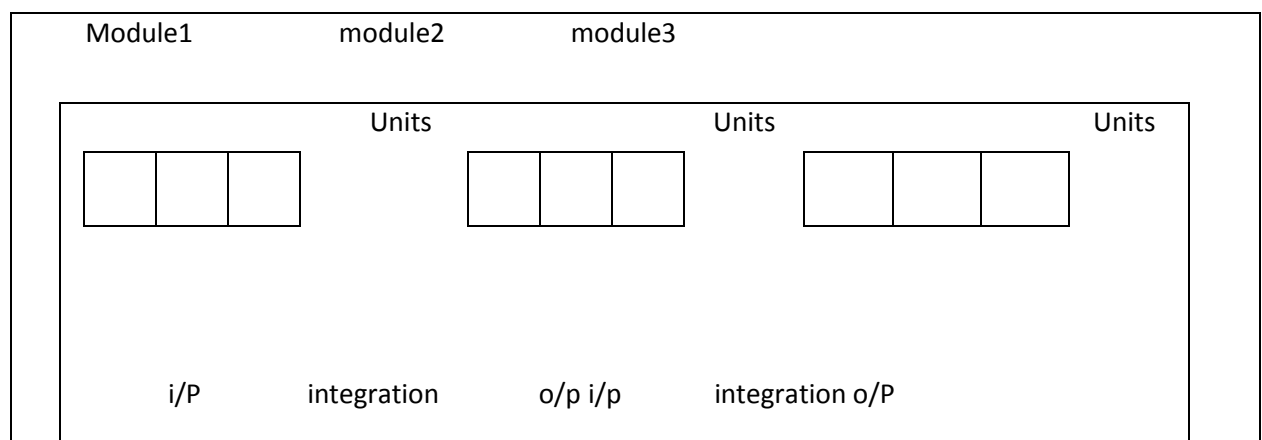
Column name	Data type	Description
Staff_id	nvarchar(50)	Id of the staff
Casual_leave	int	No of casual leaves remaining
Medical_leave	int	No of medical leaves remaining
Earn_leave	int	No of Earn leaves remaining

## 9. TESTING

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected.

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Levels of Testing:



System Testing: Presentation +business + database

UAT: user acceptance testing

### 9.1 TYPES OF TESTING:

#### Unit Testing

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two strategies:

#### Black Box Testing



In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been used to find errors in the following categories:

- a) Incorrect or missing functions
- b) Interface errors
- c) Errors in data structure or external database access
- d) Performance errors
- e) Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

### **White Box testing**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been used to generate the test cases in the following cases:

- a) Guarantee that all independent paths have been executed.
- b) Execute all logical decisions on their true and false sides.
- c) Execute all loops at their boundaries and within their operational bounds.
- d) Execute internal data structures to ensure their validity.

### **Integrating Testing**

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

### **System Testing**

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

### **Acceptance Testing**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

### **Validation**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

#### **Compilation Test:**

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

#### **Execution Test:**

This program was successfully loaded and executed. Because of good programming there was no execution error.

#### **Output Test:**

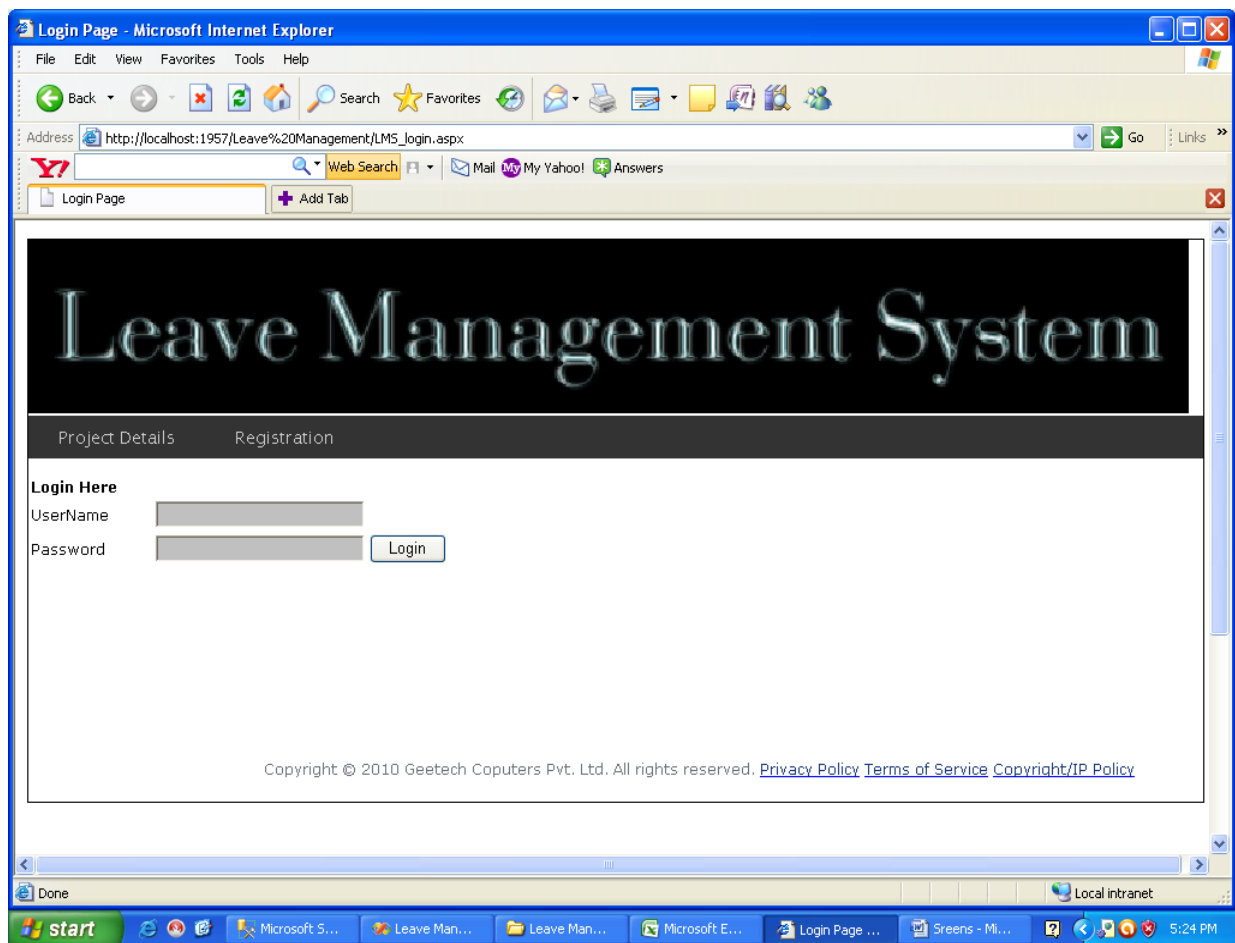
The successful output screens are placed in the output screens section above.

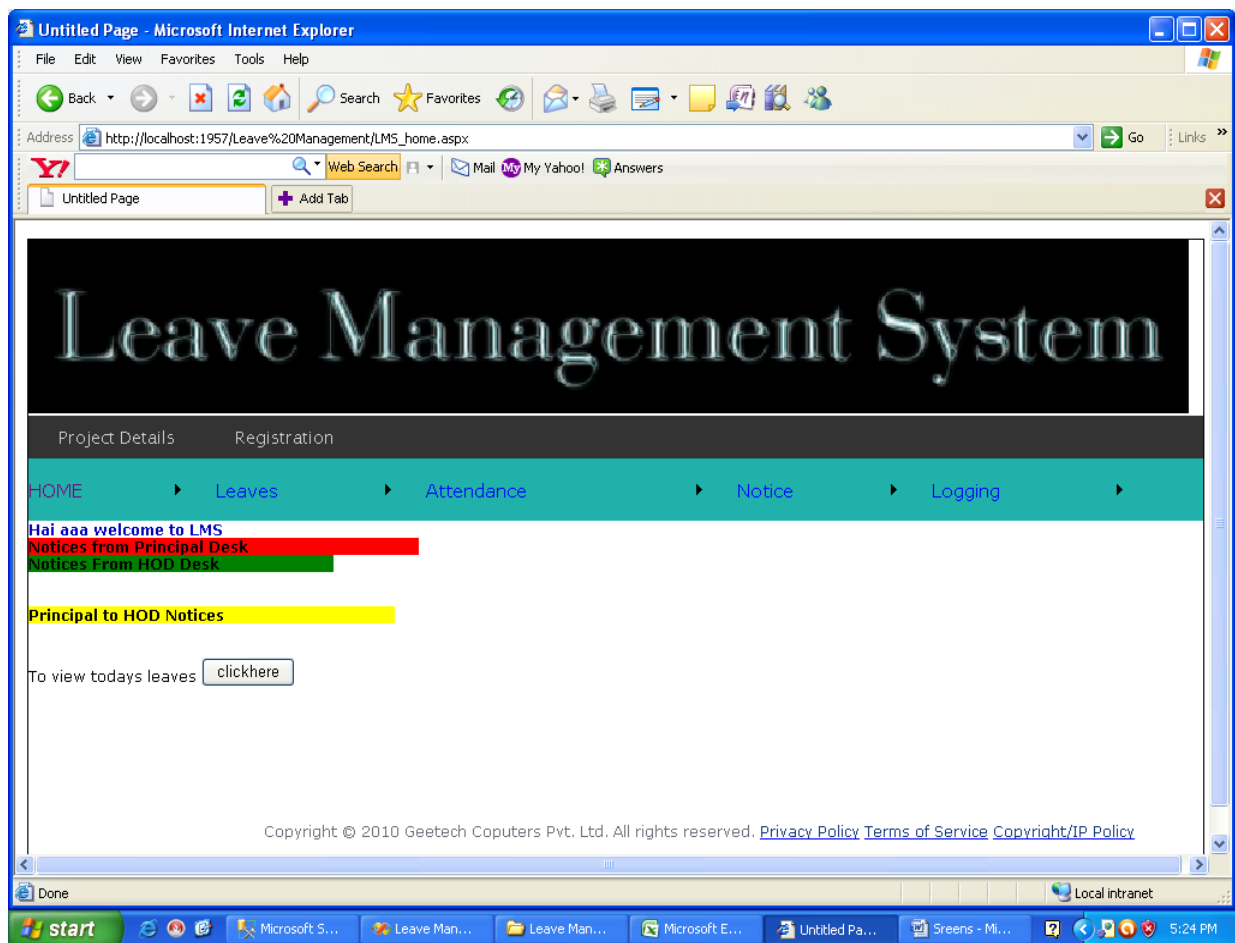
### **9.2 TEST CASES**

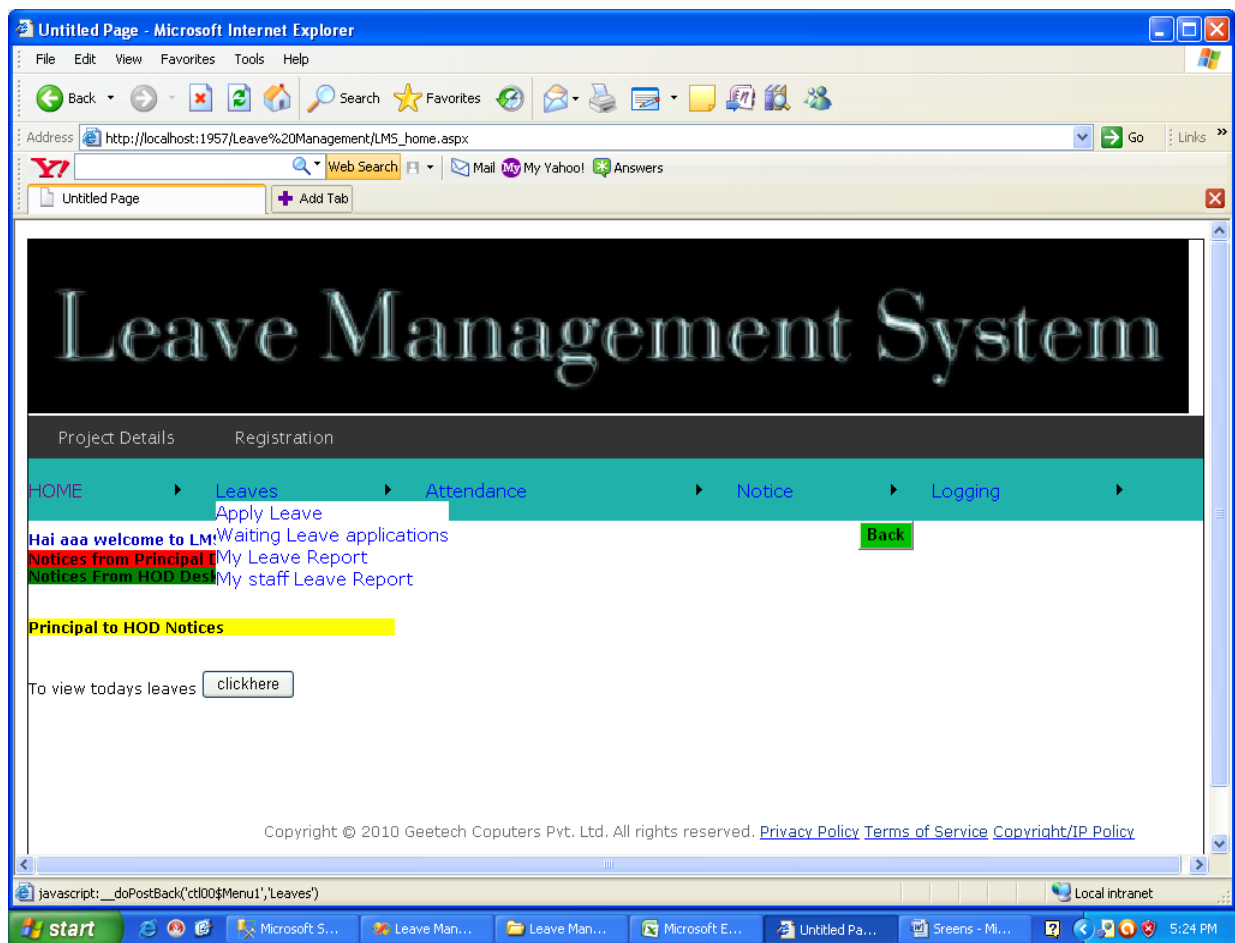
S.No.	Test Case Name	Test Procedure	Precondition	Expected Result	Output	Spec. Doc.
1	Login	Enter the Valid user name and password and click the login button	Start the Browser.	Display the home page of the respected user	Success	Test Login
2	Generate Attendance Report	In the Attendance Reports page enter the from date and to date and click generate	Start the Browser	Display the attendance records of the respected user..	Success	Test of Generate Report

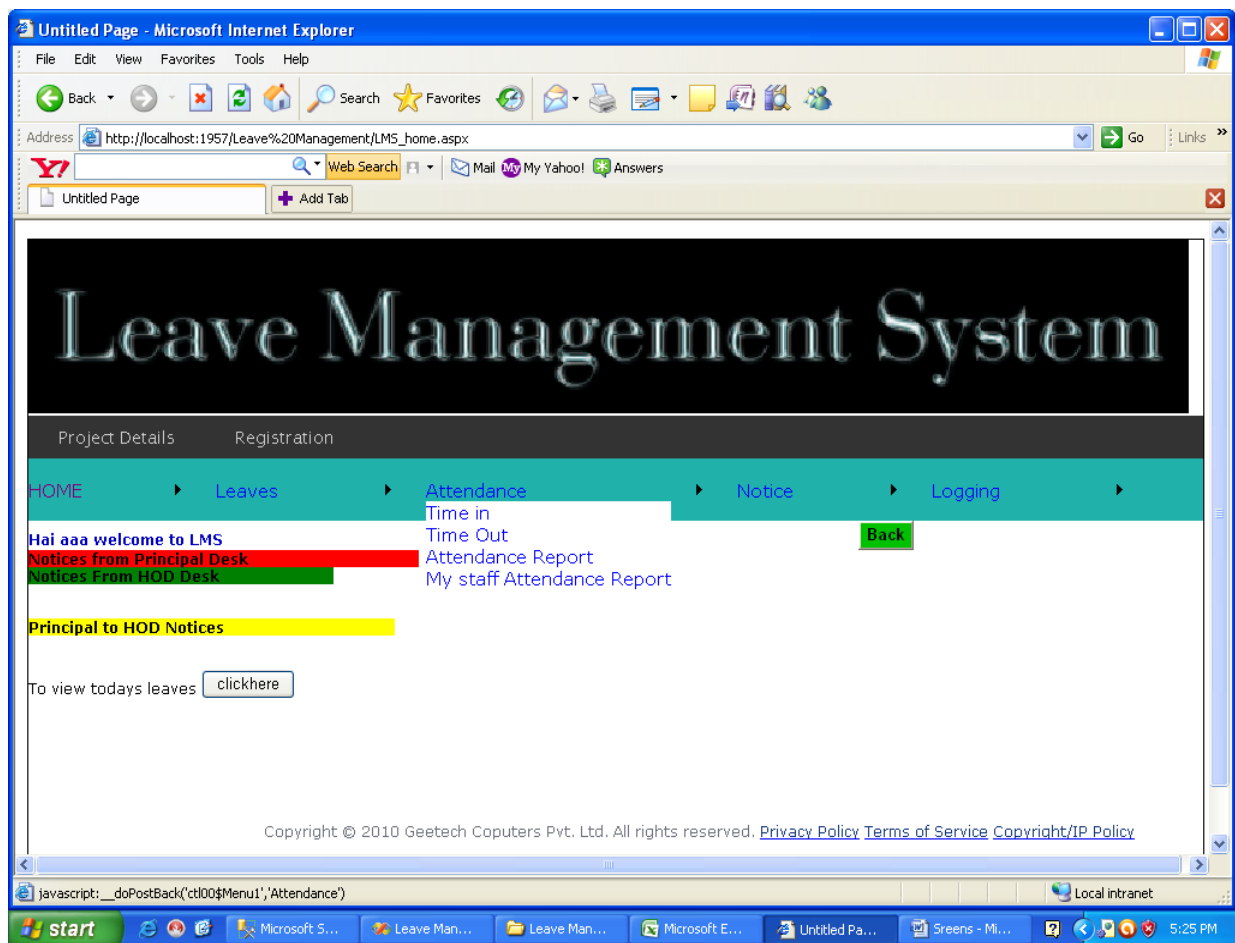
3	My Leave Report	Click the my leave report in the leaves menu item	Any leave should be applied	Display the leaves we applied and shows the remaining leaves we have	Success	Test of My leave report
4	View Notice	Click on the view notice column present in the notices menu item	Notice should be created by higher authorities.	Display all the notices and we can download those notices.	Success	Test of view notice..

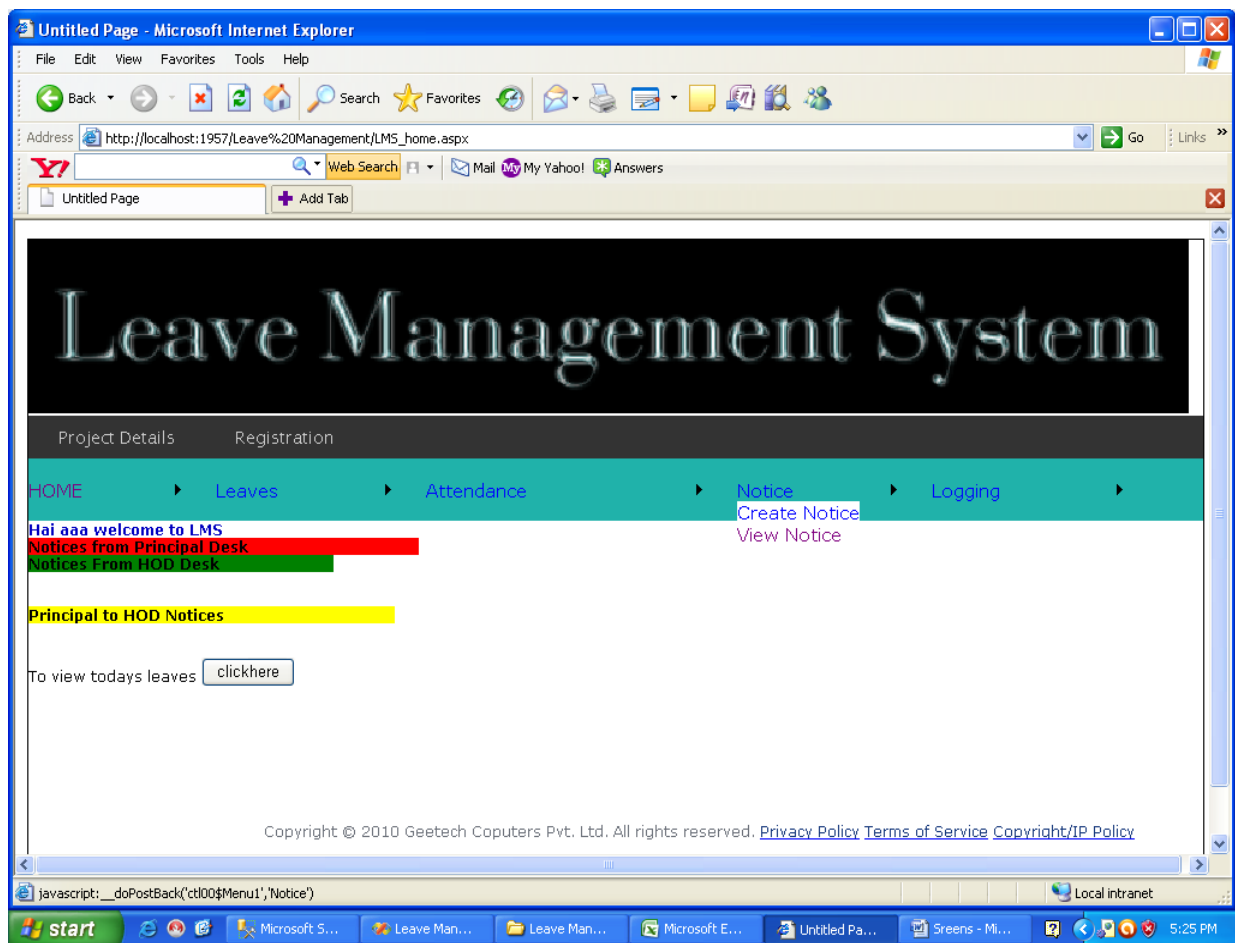
## 10. SAMPLE SCREEN SHOTS



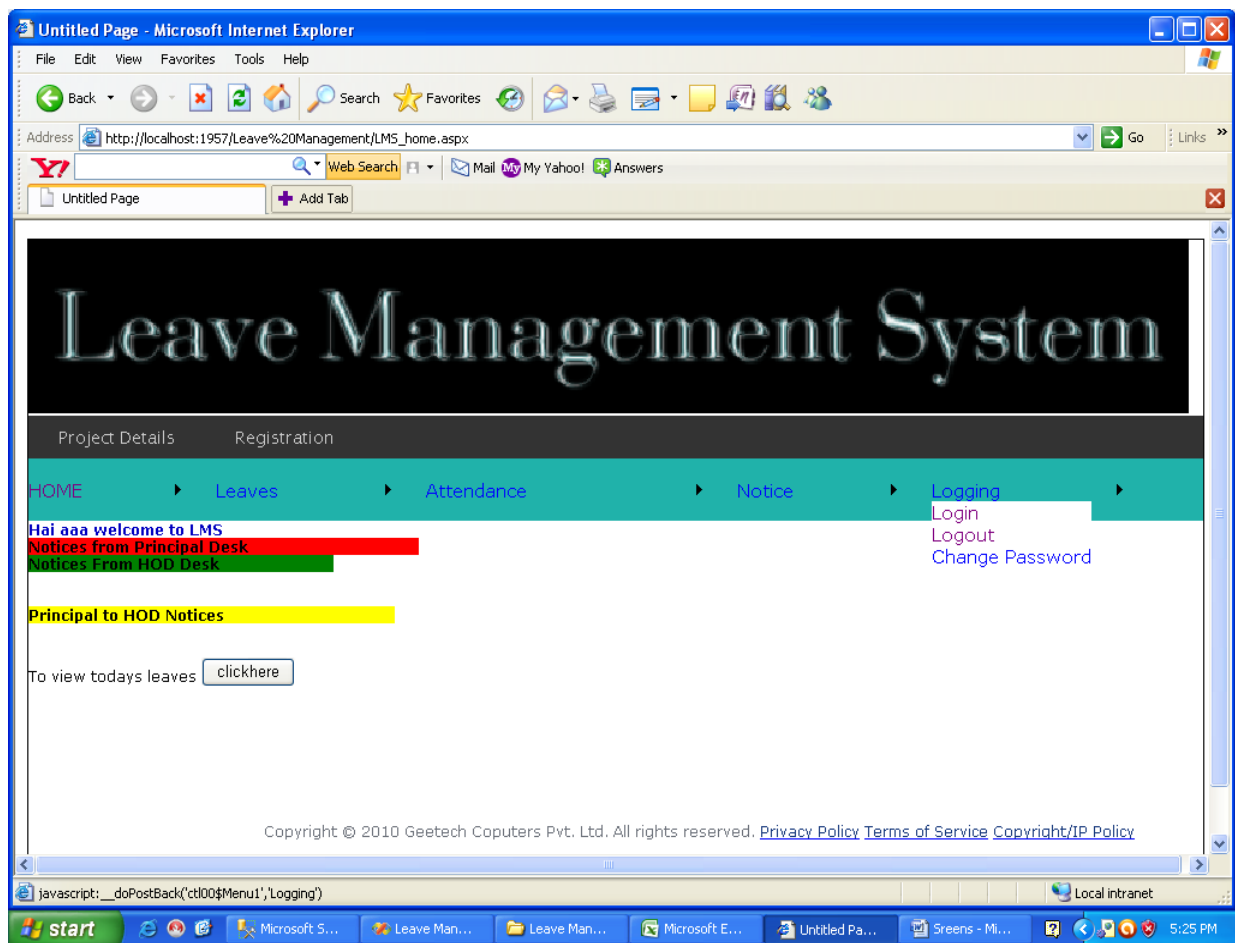












Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Mail My Yahoo! Answers

Address http://localhost:1957/Leave%20Management/LMS\_leave%20application.aspx Go Links

Web Search

Untitled Page Add Tab

Project Details Registration

HOME Leaves Attendance Notice Logging

### LEAVE APPLICATION

**Note :** (i) Leave should not be available unless sanctioned by the principal.  
(ii) Alternative arrangement for class work is to be made and same should be intimated to the class concerned in advance.  
(iii) Having signed and failed to engage the classes will entail both the faculty members to be marked absent and lead to loss of pay for the day.  
(iv) Maximum 6 Days of any type of leave can be applied

Name: sss Designation: staff

Type Of Leave: Select Number Of Days: Select

Leave StartDate: Leave EndDate: Label:

Reason:

Local intranet

start Microsoft S... Leave Man... Leave Man... Microsoft E... Screens - Mi... Untitled Pa... 5:29 PM

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Mail My Yahoo! Answers

Address http://localhost:1957/Leave%20Management/LMS\_leave%20application.aspx Go Links

Web Search

Untitled Page Add Tab

### LEAVE APPLICATION

**Note :** (i) Leave should not be available unless sanctioned by the principal.  
(ii) Alternative arrangement for class work is to be made and same should be intimated to the class concerned in advance.  
(iii) Having signed and failed to engage the classes will entail both the faculty members to be marked absent and lead to loss of pay for the day.  
(iv) Maximum 6 Days of any type of leave can be applied

Name: sss Designation: staff

Type Of Leave:  Number Of Days:

Leave StartDate: 

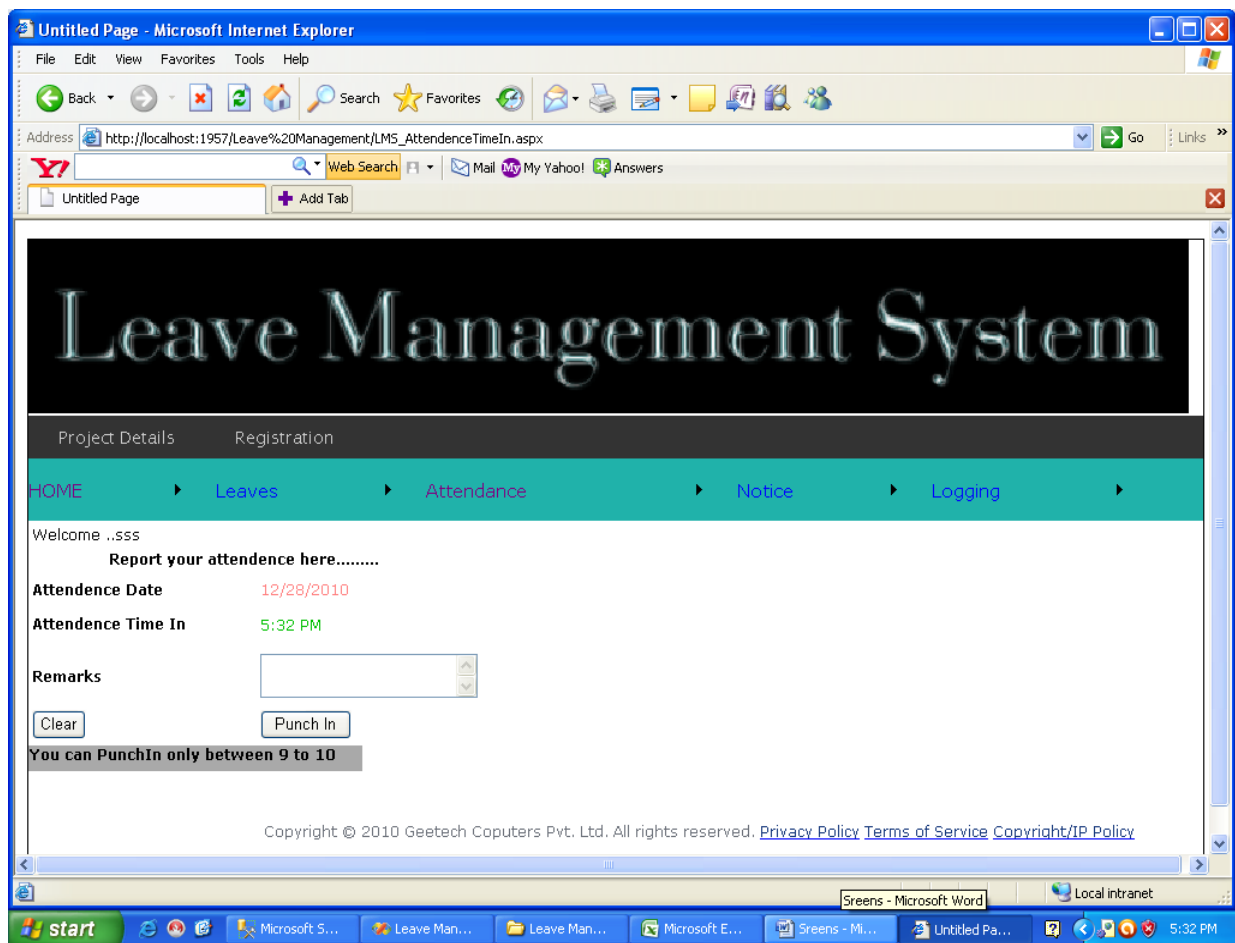
< December 2010 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

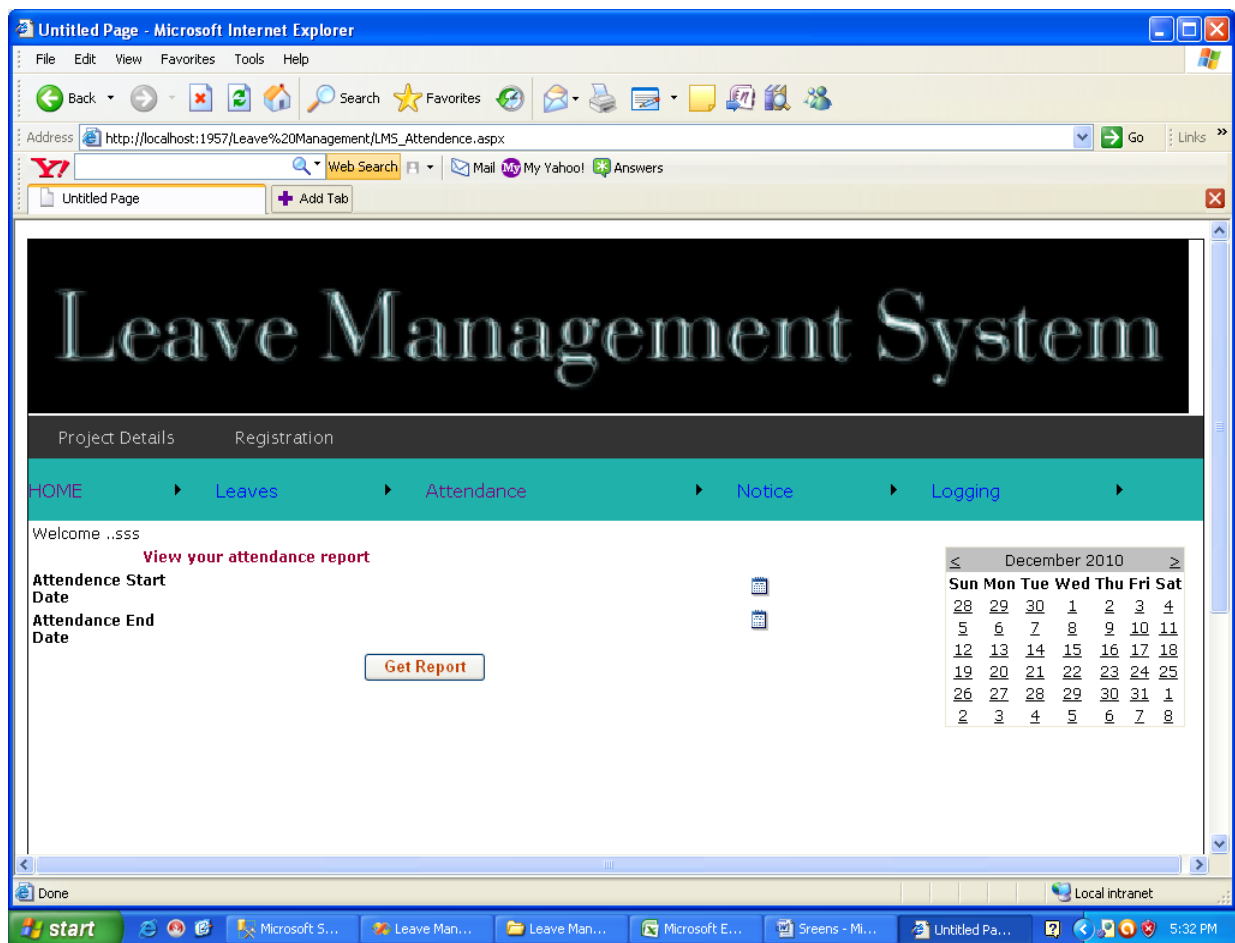
Leave EndDate: Label:

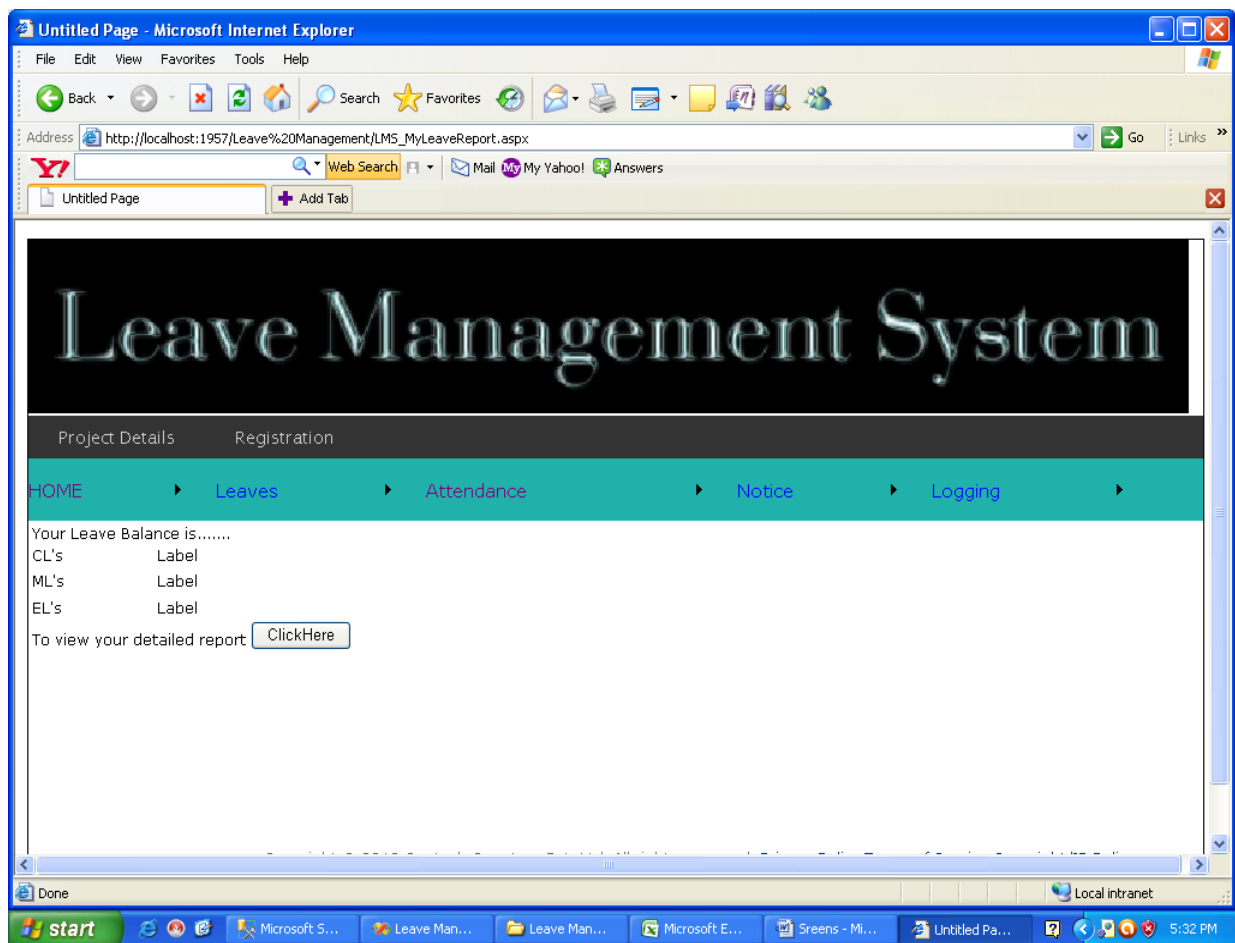
Reason:

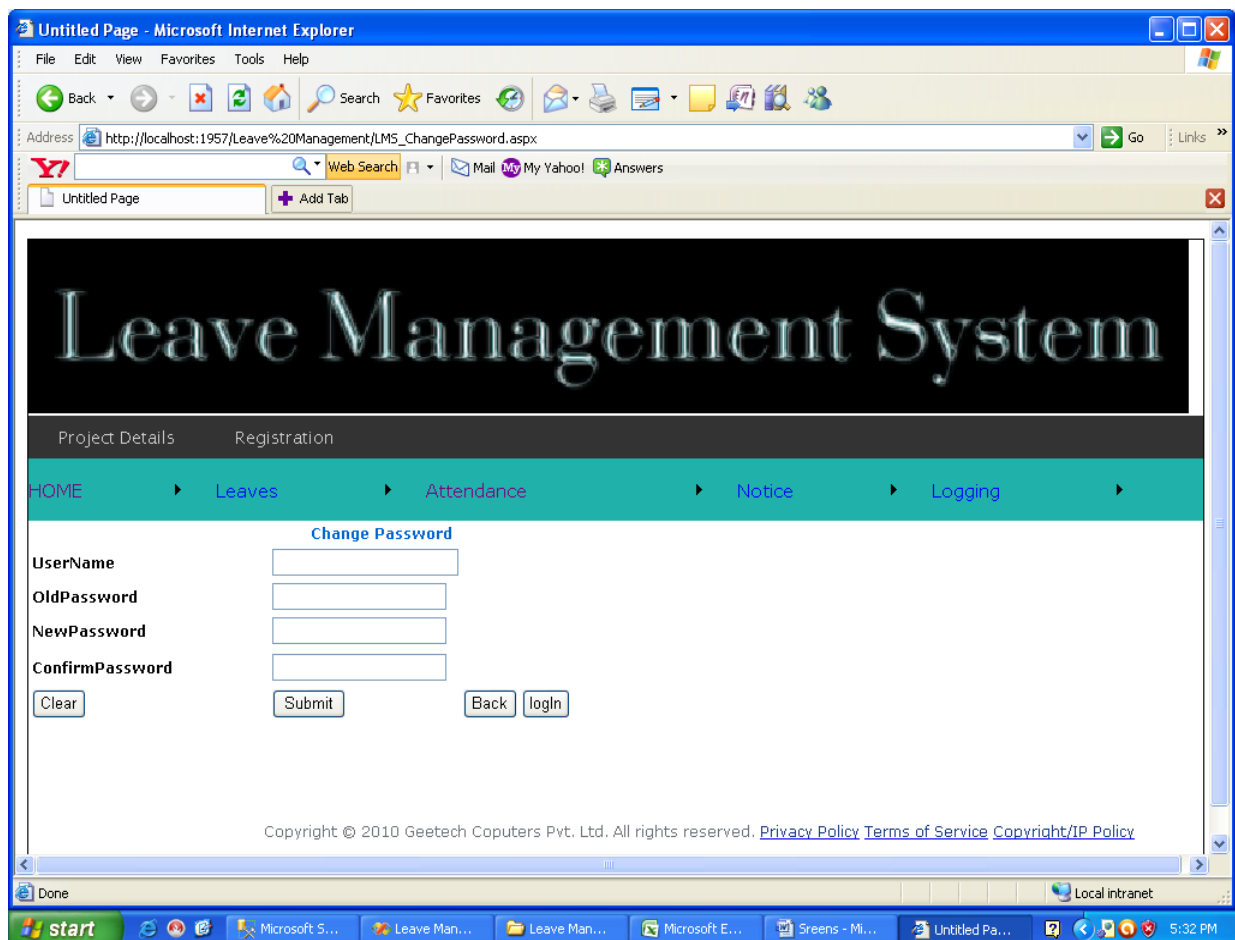
Done Local intranet

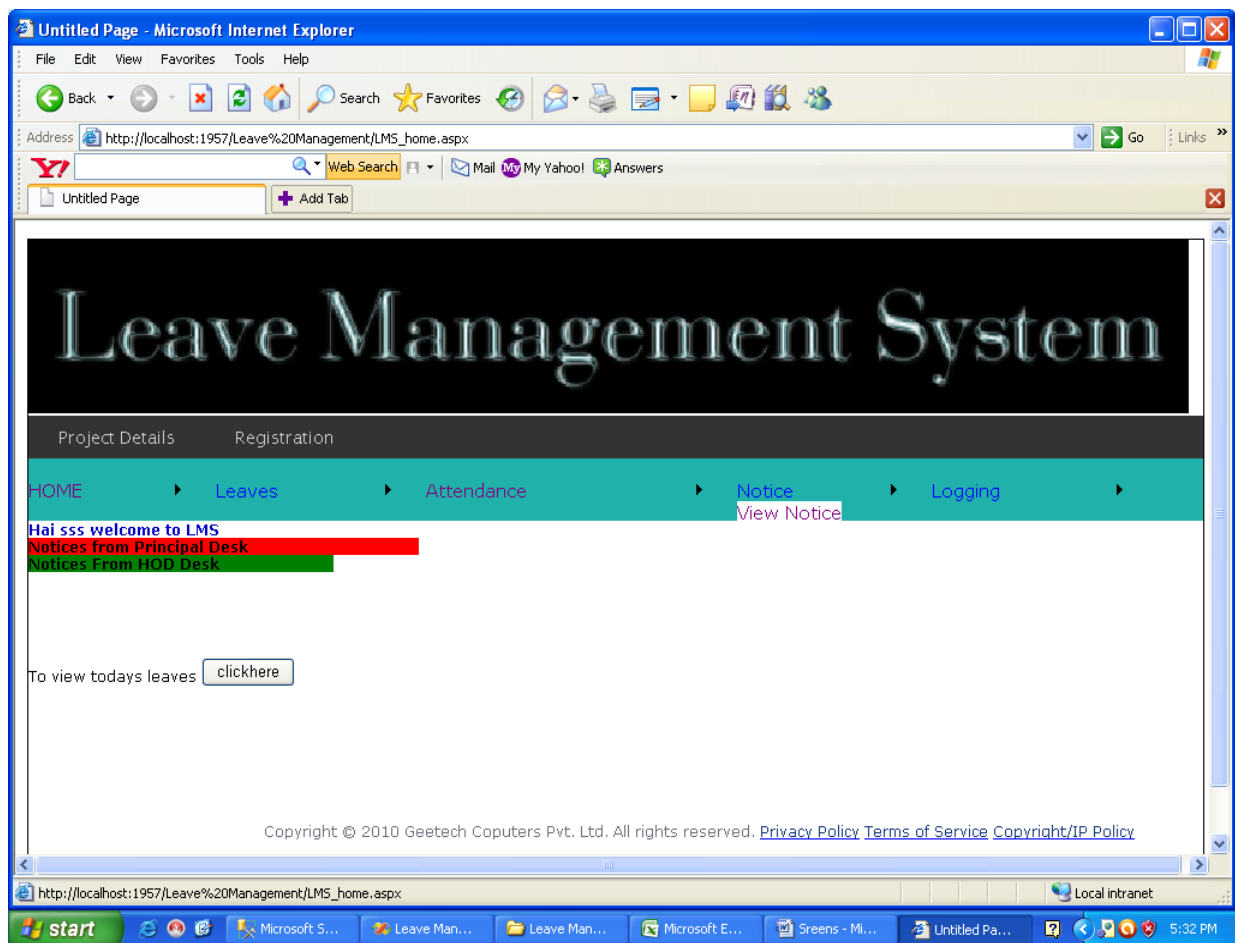
start Microsoft S... Leave Man... Leave Man... Microsoft E... Screens - Mi... Untitled Pa... 5:31 PM



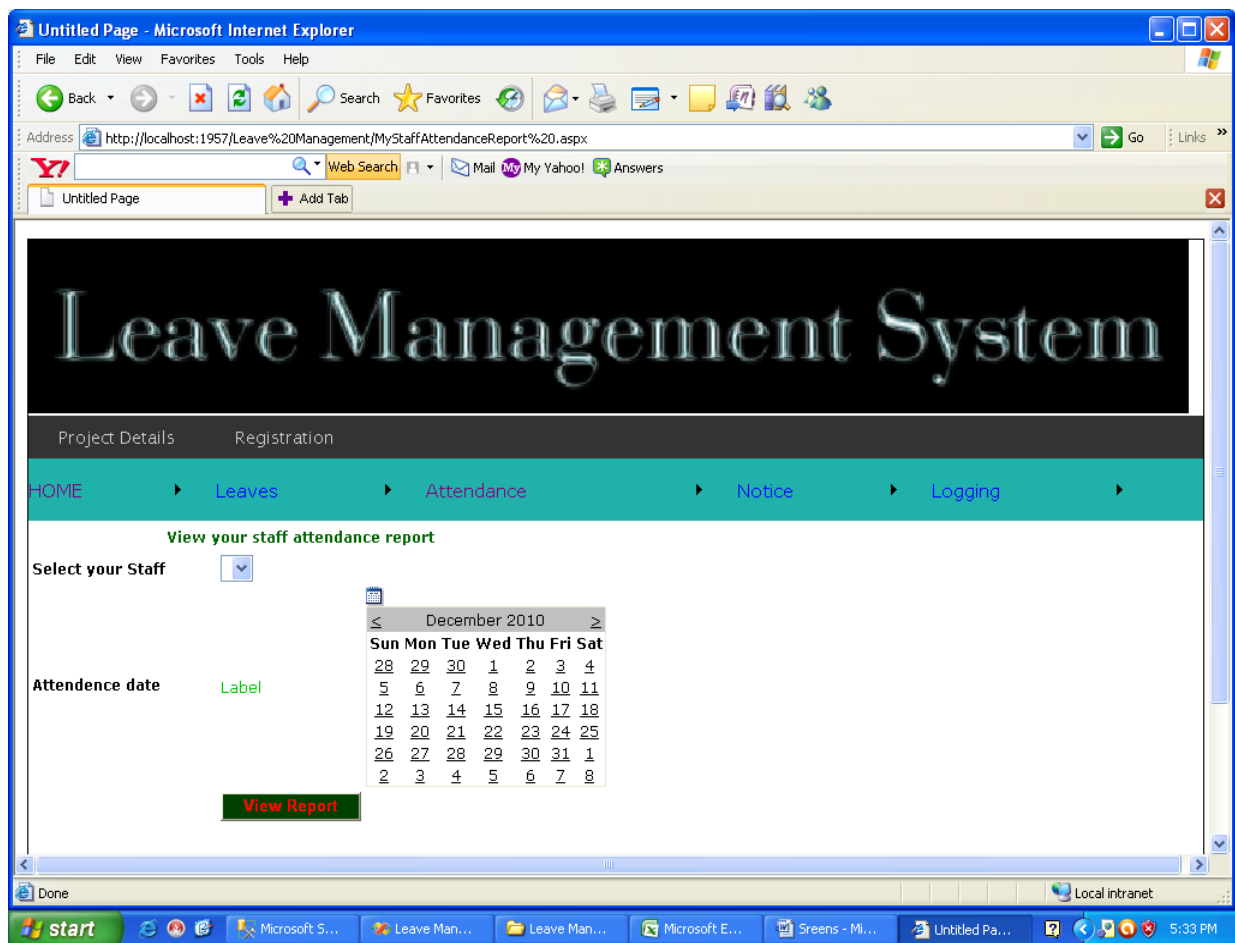


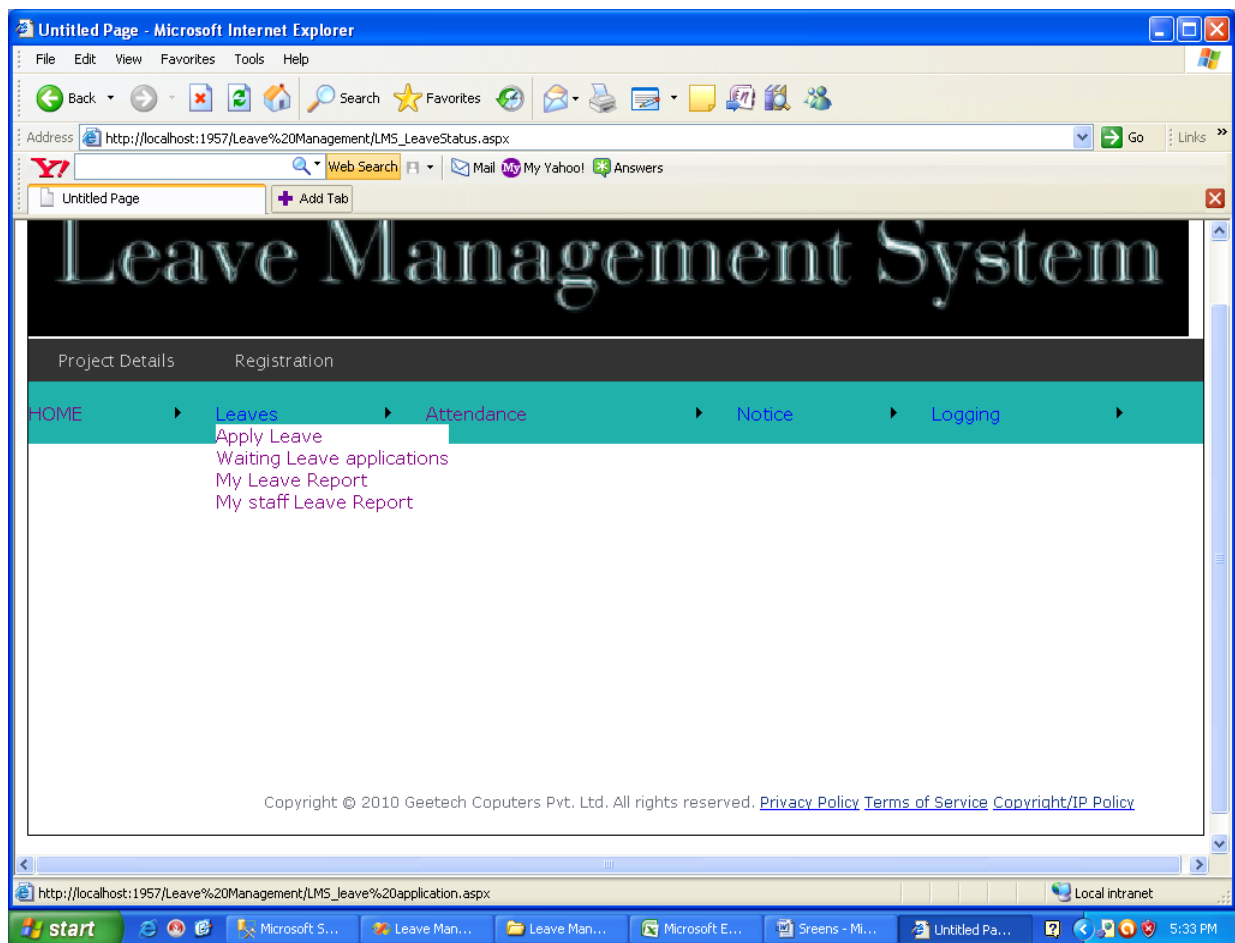


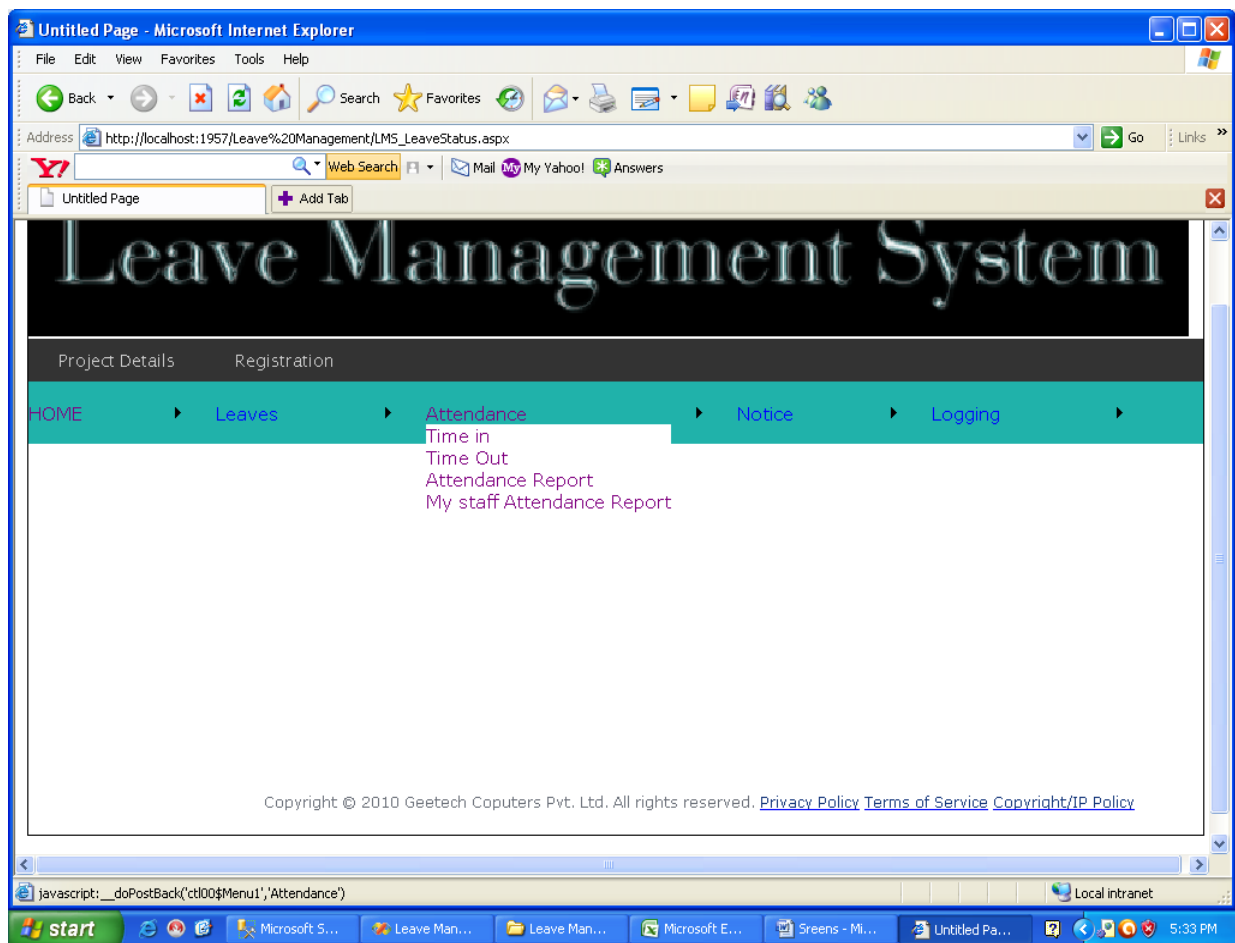


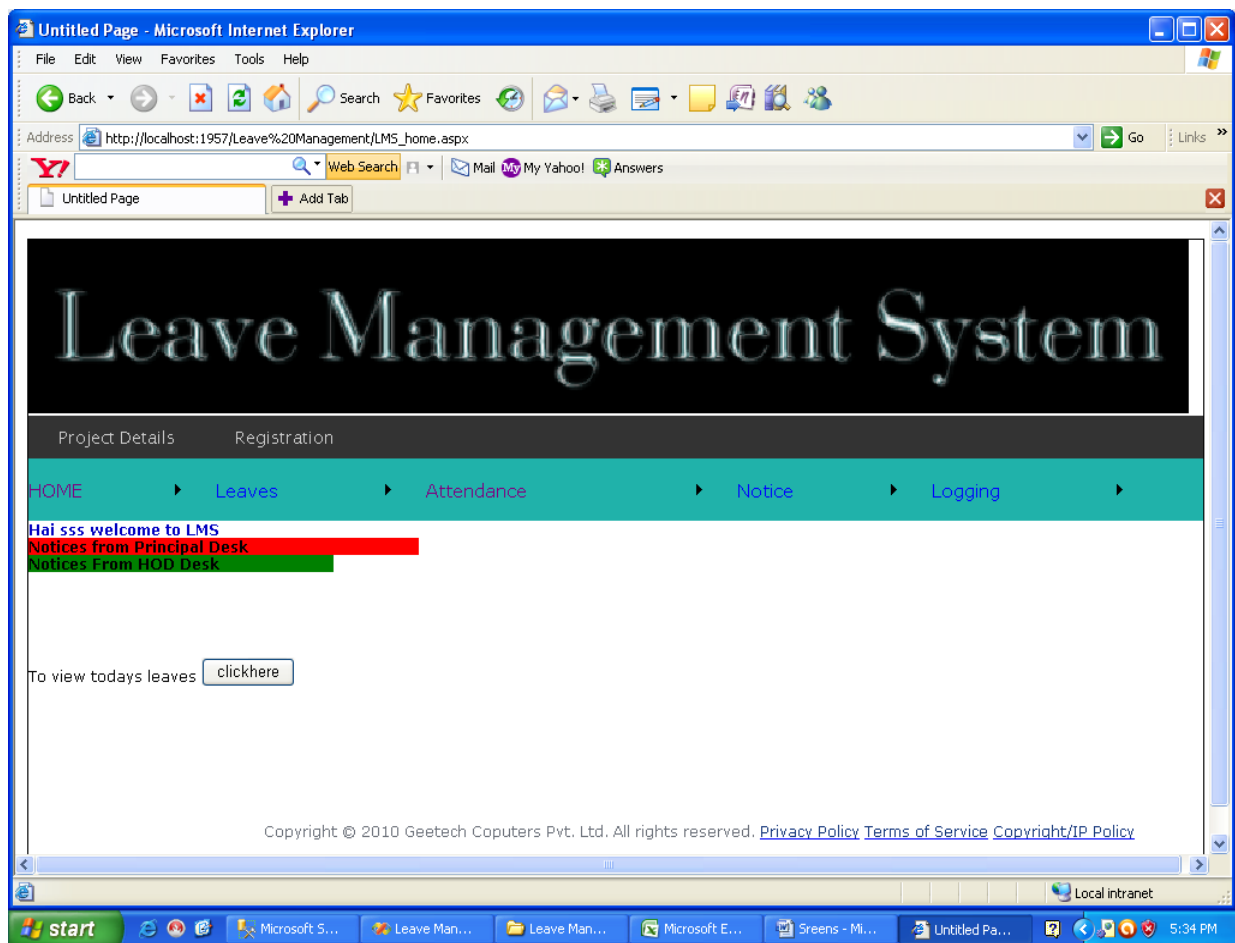


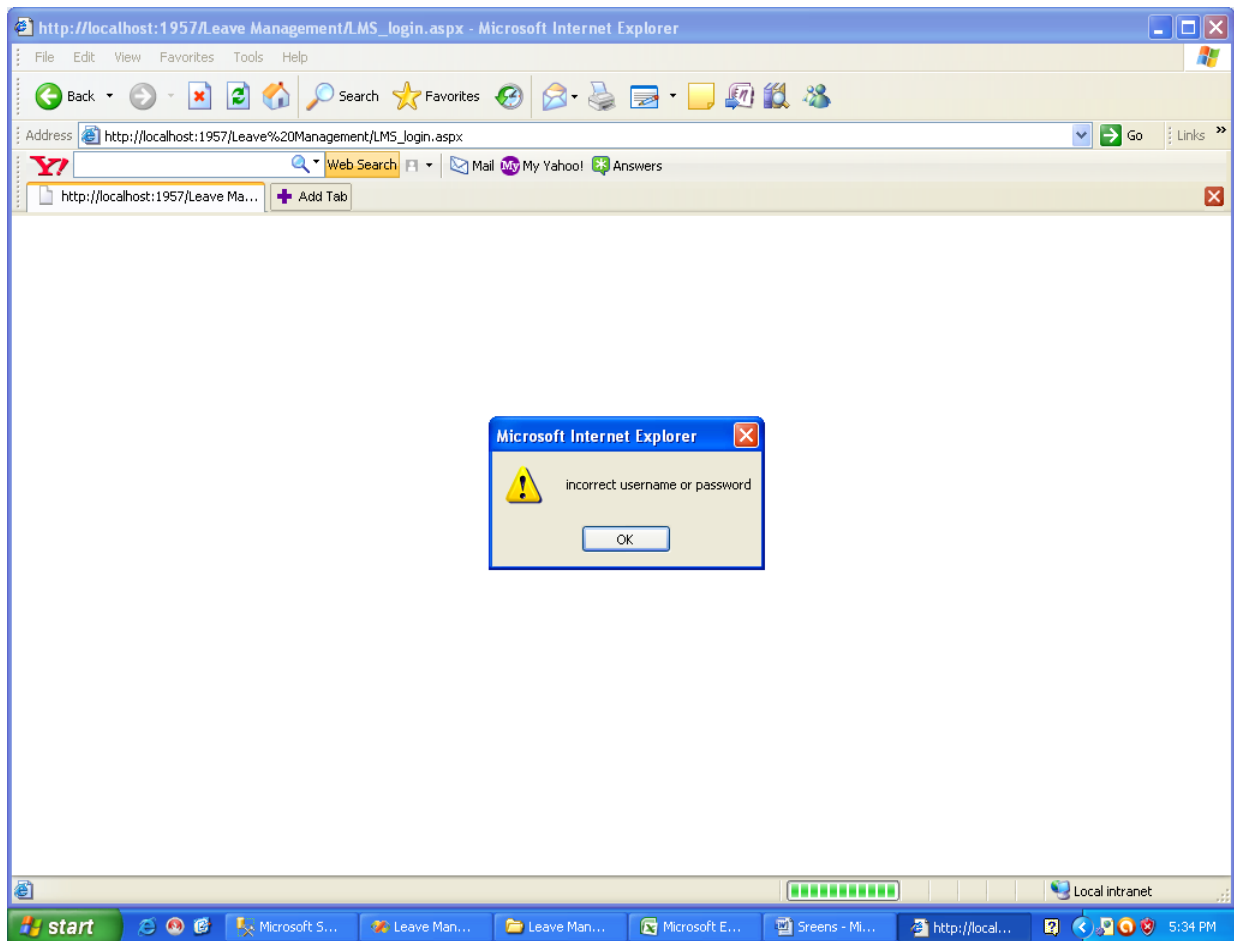












## 11. CONCLUSION & FUTURE SCOPE

### Conclusion:

Attendance Management System is very useful for college to maintain the attendance records of the staff. This system not only maintains the attendance details of the staff. it also maintains the leave applications of the staff and notices for the staff and handling of the leaves by the higher authorities

The higher authorities may accept or reject the leave applications requested by the staff. Thus this system maintains the excess amount of job done by college to maintain the attendance, leaves and notices.

This Attendance Management System is using the Login mechanism by using the Username and Password. At the time of login the user needs to Punch In and at the time of logout the user needs to Punch Out

### **Future Scope:**

Additional Features like biometric authentication while logging in can be done in the Attendance Management System. Biometric techniques like fingerprint recognition, Iris recognition are most suitable for this attendance management system. So adding the biometric authentication in place of login makes this project complete.

## **12. BIBLIOGRAPHY**

### **➤ FOR .NET INSTALLATION**

[www.support.mircosoft.com](http://www.support.mircosoft.com)

### **➤ FOR DEPLOYMENT AND PACKING ON SERVER**

[www.developer.com](http://www.developer.com)

[www.15seconds.com](http://www.15seconds.com)

### **➤ FOR SQL**

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)

➤ **FOR ASP.NET**

[www.asp.net](http://www.asp.net)

[www.asptoday.com](http://www.asptoday.com)

[www.aspfree.com](http://www.aspfree.com)

➤ **REFERENCE BOOKS**

Evangelous Petereous, *"C#.NET Black Book"*.

Roger S.Pressman, *"Software Engineering"*.

Jain, *"SQL FOR PROFESSIONALS"*.

Wrox, *"Professional ASP.NET"*.