INDUSTRIAL TRAINING REPORT

Submitted By:
Puru Govind (02bce070)
BTech (CSE)

TRAINING DETAILS

Company Name: Ebix

Start Date: 9th May'2005

End Date: 20th May'2005

Address: B59A, Sector-60
Noida-201301
U.P

Aim: To develop the attendance module of an in house project namely "E-track"

COMPANY PROFILE

Ebix stands for E-business insurance exchange. Ebix provides a comprehensive range of software solutions for insurance carrier, brokers and underwriting agencies. Our solutions are designed to meet the business requirements of a broad cross section of the market. Importantly these solutions can be tailored to meet your exact requirements.

Ebix business solutions are backed by an organization that understands the complex requirements of the market and a team of experienced business consultants and project managers with a successful track record of system implementations

Ebix leverages its **"Global Outsourcing Model"** to enable clients to take ownership of their requirement and helps them realize nearly 50% cost savings. Our delivery model encompasses multiple processes and methodologies to take the surprise out of projects and deliver measurable results for all of our global customers.

The Company has the Software Engineering Institute's **(SEI) Capability Maturity Model (CMM)** Level 5 certification for its development unit in India and has already secured an **ISO 9001:2000 certifications** for both its development and contact centre facilities in India.

History:-

A long history of designing innovative solutions has earmarked an era of excellence in the history of Ebix. It is committed in its endeavor to answer the ever-increasing needs of Insurance organizations all over the world.

The name change from "Delphi Information Systems" to Ebix Inc. aligned the identity of the company with its strategic focus of using the Internet to enhance the way insurance business is transacted, through solutions that encompass both e-commerce and web-enabled agency management systems.

An independent provider, Ebix Inc. employs hundreds of financial sector and technology professionals who provide products, support and consultancy across 6 continents.

Founded in 1976, Ebix Inc. (NASDAQ: EBIX), formerly Delphi Information Systems, is a profitable leading international supplier of software and solutions to the BFSI industry.

Ebix Inc.’s products feature fully customizable and scaleable software designed to improve the way insurance professionals manage all aspects of BFSI distribution, including: marketing, sales, service, accounting and management.

Both the analyst and the media community have recognized Ebix for its innovation and technology expertise.

TECHNOLOGIES USED

Microsoft .NET

The **Microsoft .NET Framework**, more commonly known as simply the **.NET Framework**, is a software development platform created by Microsoft. The .NET Framework is now in version 2.0, which was released in October of 2005 and is the successor to two major previous versions: 1.0 and 1.1.

Description

The .NET Framework was designed with several intentions:

Interoperability - Because so many COM libraries have already been created, the .NET Framework provides methods for allowing interoperability between new code and existing libraries.

Common Runtime Engine - Like the Java platform, programming languages on the .NET Framework compile into an intermediate language known as the Common Intermediate Language, or CIL; Microsoft's implementation of CIL is known as Microsoft Intermediate Language, or MSIL. Unlike the Java platform, however, this intermediate language is not interpreted, but rather compiled in a manner known as just-in-time compilation (JIT) into native code. The combination of these concepts is called the Common Language Infrastructure (CLI), a specification; Microsoft's implementation of the CLI is known as the Common Language Runtime (CLR).

Language Independence - Unlike the Java platform and COM, the .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other. Because of this feature, the .NET Framework supports development in multiple programming languages. This is discussed in more detail in the .NET languages section below.

Base Class Library - The Base Class Library (BCL), sometimes referred to as the Framework Class Library (FCL), is a library of types available to all languages using the .NET Framework. The BCL provides classes which encapsulate a number of common functions such as file reading and writing, graphic rendering, database interaction, XML document manipulation, and so forth.

Simplified Deployment - Installation and deployment of Windows applications has been the bane of many developers' existence. Registry settings, file distribution and DLL hell

have been nearly completely eliminated by new deployment mechanisms in the .NET Framework.

Security - As the Internet becomes more and more integrated into our computing lives, the issue of what code should be trusted comes of high importance. The .NET Framework introduces code access security to address this issue.

Because of the inherent nature of the .NET Framework, through its use of an intermediate language, it is platform independent. While the consensus is that the .NET Framework is only available for the Microsoft Windows operating system, this is innaccurate. Microsoft provides the Shared Source Common Language Infrastructure, a version of the CLI, for Windows, FreeBSD and Mac OS X. Additionally, because the CLI is now an Ecma International standard, several open source development projects have sprung up to provide support for additional platforms; the most notable of these projects are Mono, DotGNU and Portable.NET. Microsoft also provides a scaled-down version of the .NET Framework for use on smart devices, such as Pocket PCs and Smartphones, called the .NET Compact Framework.

Although the .NET Framework is available and provides compilers at no cost, Microsoft does offer a number of additional tools for easing the development process. The most prominent tool is the Visual Studio .NET integrated development environment. There is also an open source IDE known as SharpDevelop.

History

Though some believe that technologies used in .NET were originally developed by Microsoft as their version of the Java platform, the truth is that many of the teams working on .NET initially set out to create COM+ 2.5. Other departments were also improving other Microsoft technologies; the web server department was creating ASP 4.0 and the Microsoft distributed computing departments were creating what was called "Next-Generation Web Services". The work from the various departments merged into one system now called .NET.

When Microsoft decided to end their future use of Sun's Java technologies in 1998, the existing Microsoft J++ (Java) product was transformed into the beginnings of the .NET project. Code from the .NET Common Language Runtime (CLR) was said to have come from Colusa Software's OmniVM, which Microsoft acquired on March 12, 1996. Despite the rumors, the Microsoft CLR is not truly comparable to the Java VM as the CLR is a common *language* runtime, not a common *platform* runtime as the Java VM is. The CLR is actually the Microsoft implementation of the ECMA CIL standard. While Java applications can be written for many different platforms in only one language, .NET applications can be written in many different languages for, currently, only one operating-system platform (though many hardware platforms).

While the original model of .NET was that of a general foundation (.NET framework) with three primary pillars (ASP.NET, Windows, and Web Services), the model for .NET

2.0 is that of a foundation for Microsoft's next generation platform known as WinFX, which is the unification of Microsoft development technologies into one programming model. WinFX is also the replacement for the longlived Win32 API introduced in the early 1990s. The FX in WinFX is the abbreviation for the .NET Framework thus giving .NET the tall standing of being the foundation of the majority of future Microsoft-based development, higher or lower level.

LANGUAGES AND SCRIPTING LANGUAGES USED

SERVER SIDE PROCESSING(ASP.net and C#)

ASP.NET is a set of web development technologies marketed by Microsoft. Programmers can use it to build dynamic web sites, web applications and XML web services. It is part of Microsoft's .NET platform and is the successor to Microsoft's Active Server Pages (ASP) technology

Principles of ASP.NET

Even though ASP.NET takes its name from Microsoft's old web development technology, ASP, the two differ significantly. Microsoft has completely rebuilt ASP.NET, based on the Common Language Runtime (CLR) shared by all Microsoft .NET applications. Programmers can write ASP.NET code using any of the different programming languages supported by the .NET framework, usually (proprietary) Visual Basic.NET, JScript .NET, or (standardized) C#, but also including open-source languages such as Perl and Python. ASP.NET is faster because the entire web site is precompiled to one or a few DLL files on a web server and the web site runs faster compared to the previous scripting technology.

ASP.NET attempts to simplify developers' transition from Windows application development to web development by allowing them to build pages composed of *controls* similar to a Windows user interface. A web control, such as a *button* or *label*, functions in very much the same way as its Windows counterpart: code can assign its properties and respond to its events. Controls know how to render themselves: whereas Windows controls draw themselves to the screen, web controls produce segments of HTML which form part of the resulting page sent to the end-user's browser.

ASP.NET encourages the programmer to develop applications using an event-driven GUI paradigm, rather than in the conventional web scripting fashion. The framework attempts to combine existing technologies such as JavaScript with internal components like "Viewstate" to bring persistent (inter-request) state to the inherently stateless web environment.

ASP.NET uses the .NET Framework as an infrastructure. The .NET Framework offers a managed runtime environment (like Java), providing a virtual machine with JIT and a class library.

The numerous .NET controls, classes and tools can cut down on development time by providing a rich set of features for common programming tasks. Data access provides one example, and comes tightly coupled with ASP.NET. A developer can make a page to display a list of records in a database, for example, significantly more readily using ASP.NET than with ASP.

Advantages of ASP.NET over ASP

- Compiled code means applications run faster with more design-time errors trapped at the development stage
- Significantly improved run-time error handling, making use of exceptions and Try-Catch blocks.
- User-defined controls allow commonly used templates, such as menus
- Similar metaphors to Windows applications such as controls and events, which make development of rich user interfaces, previously only found on the desktop, possible.
- A rich set of controls and class libraries allows the rapid building of applications
- ASP.NET leverages the multi language capabilities of the .NET CLR, allowing web pages to be coded in VB.NET, C#, J#, etc.
- Ability to cache the whole page or just parts of it to improve performance.
- Ability to use the CodeBehind development model to separate business logic from presentation.
- If an ASP.NET application leaks memory, the ASP.NET runtime unloads the AppDomain hosting the erring application and reloads the application in a new AppDomain.
- Session state in ASP.NET can be saved in a SQL Server database or in a separate process running on the same machine as the web server or on a different machine. That way session values are not lost when IIS is reset or the ASP.NET worker process is recycled.

Disadvantages to other platforms

- The server framework runs natively on Microsoft IIS 5.0 or higher and Cassini, a web server developed in .NET (shipped with WebMatrix, a free ASP.NET 1.1 development environment, and Visual Studio 2005); however it can run on Linux on any of the alternative frameworks based on the ECMA standard. The most well known one is Mono Project, a free/opensource framework.
- Previous versions of ASP.NET (1.0 and 1.1) were criticized for their lack of standards compliance. The generated HTML and JavaScript sent to the client browser would not always validate against W3C/ECMA standards. In addition, the framework's browser detection feature sometimes incorrectly identified web browsers other than Microsoft's own Internet Explorer as "downlevel" and

returned HTML/JavaScript to these clients that was crippled or broken. However, in version 2.0, all controls generate valid HTML 4.0, XHTML 1.0 (the default), or XHTML 1.1 output, depending on the site configuration, detection of standards-compliant web browsers is more robust, and support for Cascading Style Sheets is more extensive.

Development tools

Several available software packages exist for developing ASP.NET applications:

- Visual Studio .NET (Proprietary)
- ASP.NET Web Matrix (Proprietary)
- Macromedia Dreamweaver MX 2004 (Proprietary)
- Macromedia Dreamweaver 8 (Proprietary)
- Macromedia HomeSite 5.5 (For ASP Tags) (Proprietary)
- Microsoft SharePoint Designer 12 (Proprietary)
- Visual Web Developer 2005 Express Edition (for ASP.NET 2.0) (Proprietary)
- Delphi 2005 (Proprietary)
- SharpDevelop (Free/Open Source)

Current version

On November 7th, 2005, Microsoft released ASP.NET version 2.0 as part of the next version of the .NET Framework. At the same time, a new version of its software development tool was released (Visual Studio 2005) as well as its database software (SQL Server 2005).

ASP.NET Team Members

Various ASP.NET team members maintain blogs. Here are some of them:

- Scott Guthrie, Product Unit Manager - http://weblogs.asp.net/scottgu
- Nikhil Kothari, Architect - http://www.nikhilk.net
- Brian Goldfarb, Product Manager - http://blogs.msdn.com/bgold

C# (see section on naming, pronunciation) is an object-oriented programming language developed by Microsoft as part of their .NET initiative. Microsoft based C# on C++ and Java. C# has a syntax that combines aspects of many programming languages. Some notable contributions are from C++, Java and Delphi. Other contributions can be traced on a programming language history chart. C# fits well for rapid development, like Visual Basic or Delphi.

Standardization

Microsoft has submitted C# to the ECMA for formal standardization. In December 2001, ECMA released ECMA-334 *C# Language Specification*. C# became an ISO standard in 2003 (ISO/IEC 23270). There are independent implementations being worked on, including:

- Mono, Novell's open source .NET implementation (originally by Ximian).
- dotGNU, and Portable.NET from the Free Software Foundation
- More recently, Microsoft has added support in release of Visual Studio 2005 for generics (similar to C++ templates), partial types and some other new features. ECMA/ISO standardization of these new features has been proposed, but they are not currently part of the standard language definition.

Politics

Many of Microsoft's products and initiatives generate political attention, and C# is no exception. Owing to C#'s close relationship with a commercial institution, political discussions continue regarding the legitimacy of C# standardization, its Java similarities, its future as a general-purpose language, and other issues. Some security experts express skepticism as to the efficacy of the CLR's security mechanisms, and criticise their complexity. At the same time, the language is praised for its clear and programmer-friendly grammar, in addition to reduction in development time for certain types of applications.

Unlike proprietary languages such as Visual Basic or Java, Microsoft chose to open up C# to the standardization process. However, Microsoft is still a primary force driving changes and innovation in the language. Additionally, Microsoft has made it clear that C#, as well as the other .NET languages, is an important part of its software strategy for both internal use and external consumption. Microsoft takes an active role in marketing the language as part of its overall business strategies.

Language name

According to the ECMA-334 C# Language Specification, section 6, *Acronyms and abbreviations* [1] the name of the language is written "C#" ("LATIN CAPITAL LETTER C (U+0043) followed by the NUMBER SIGN # (U+0023)") and pronounced "C Sharp".

The name "C#" may have been chosen by Microsoft to imply progression from the C++ language, with the # symbol resembling two ++ symbols merged together, or four + symbols arranged in a square. However, it may also imply that it is not as far from C as C++, since ++ signifies incrementing (adding one whole) whereas the musical sharp symbol signifies the note one half step above another note, showing that C# is only a half step above C.

C sharp musical note

Due to technical limitations of display (fonts, browsers, etc.) and the fact that the sharp symbol (♯, **U+266F, MUSIC SHARP SIGN**, see graphic at right if the symbol is not visible) is not present on the standard keyboard, the number sign (#) was chosen to represent the sharp symbol in the written name of the language. So, although the symbol in "C#" represents the sharp symbol, it is actually the number sign ("#"). Although Microsoft's C# FAQ refers to the sharp symbol in the language name, Microsoft clarifies the language name as follows:

"The spoken name of the language is "C sharp" in reference to the musical "sharp" sign, which increases a tone denoted by a letter (between A and G) by half a tone. However, for ease of typing it was decided to represent the sharp sign by a pound symbol (which is on any keyboard) rather than the "musically correct" Unicode sharp sign. The Microsoft and ECMA 334 representation symbols thus agree: the # in C# is the pound sign, but it represents a sharp sign. Think of it in the same way as the <= glyph in C languages which is a less than sign and an equals sign, but represents a less-than-or-equals sign.", Microsoft Online Customer Service

The choice to represent the sharp symbol (♯) with the number sign (#) has led to confusion regarding the name of the language. For example, although most printed literature uses the correct number sign [2], some incorrectly uses the sharp symbol. What's more, users have been known to call the language "*see-pound*" (in the US the #-key on telephones is pronounced as the "*pound*"-key) or "*see-hash*". Also in the US the # symbol is also occasionally referred to as the "*gate*" symbol on a telephone, leading to a pronunciation of the language as "*see-gate*", which could be confused with the brand name of hard-drive manufacturer, Seagate.

The "sharp" suffix has been emulated by a number of other .NET languages that are variants of existing languages, including J# (Microsoft's implementation of Java), A# (from Ada), F# (presumably from System F, the type system used by the ML family), and Gtk# (a .NET wrapper for GTK+).

CLIENT SIDE PROCEESING(JAVASCRIPT)

JavaScript is an object-based scripting programming language based on the concept of prototypes. The language is best known for its use in websites, but is also used to enable scripting access to objects embedded in other applications.

Despite the name, JavaScript is only distantly related to the Java programming language, the main similarity being their common debt to the C programming language. JavaScript has far more in common with the Self programming language.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

History

JavaScript was originally developed by Brendan Eich of Netscape Communications Corporation under the name *Mocha*, then *LiveScript*, and finally renamed to JavaScript. The change of name from LiveScript to JavaScript roughly coincided with Netscape adding support for Java technology in its Netscape Navigator web browser. JavaScript was first introduced and deployed in the Netscape browser version 2.0B3 in December of 1995. When web developers talk about using JavaScript in Internet Explorer, they are actually using JScript. The choice of name proved to be a source of much confusion.

As of 2005, the latest version of the language is JavaScript 1.6, which corresponds to ECMA-262 Edition 3 like JavaScript 1.5, except for Array extras, and Array and String generics. ECMAScript, in simple terms, is a standardized version of JavaScript. Mozilla versions since 1.8 Beta 1 also have partial support of E4X, which is a language extension dealing with XML, defined in the ECMA-357 standard.

Usage

JavaScript is a prototype-based scripting language with a syntax loosely based on C. Like C, it has the concept of *reserved keywords*, which (being executed from source) means it is almost impossible to extend the language without breakage.

Also like C, the language has no input or output constructs of its own. Where C relies on standard I/O libraries, a JavaScript engine relies on a *host environment* into which it is embedded. There are many such host environment applications, of which web technologies are the most well known examples. These are examined first.

JavaScript embedded in a web browser connects through interfaces called Document Object Model (DOM) to applications, especially to the server side (web servers) and the client side (web browsers) of web applications. Many web sites use client-side JavaScript

technology to create powerful dynamic web applications. It may use unicode and can evaluate regular expressions (introduced in version 1.2 in Netscape Navigator 4 and Internet Explorer 4). JavaScript expressions contained in a string can be evaluated using the `eval` function.

One major use of web-based JavaScript is to write functions that are embedded in or included from HTML pages and interact with the DOM of the page to perform tasks not possible in static HTML alone, such as opening a new window, checking input values, changing images as the mouse cursor moves over, etc. Unfortunately, the DOM interfaces in various browsers differ and don't always match the W3C DOM standards. Different browsers expose different objects and methods to the script. It is therefore often necessary to write different variants of a JavaScript function for the various browsers, though this situation is improving. Major design methodologies using JavaScript to interact with DOM include DHTML, Ajax, and SPA.

Outside of the Web, JavaScript interpreters are embedded in a number of tools. Adobe Acrobat and Adobe Reader support JavaScript in PDF files. The Mozilla platform, which underlies several common web browsers, uses JavaScript to implement the user interface and transaction logic of its various products. JavaScript interpreters are also embedded in proprietary applications that lack scriptable interfaces. Dashboard Widgets in Apple's Mac OS X v10.4 are implemented using JavaScript. Microsoft's Active Scripting technology supports JavaScript-compatible JScript as an operating system scripting language. JScript .NET is a CLI-compliant language that is similar to JScript, but has further object oriented programming features.

Each of these applications provides its own object model which provides access to the host environment, with the core JavaScript language remaining mostly the same in each application.

Debugging

Depending on the development environment debugging used to be difficult. Since errors in JavaScript only appear in run-time (i.e., there is no way to check for errors without executing the code), and since JavaScript is interpreted by the web browser as the page is viewed, it may be difficult to track the cause for errors. However nowadays both Internet Explorer and the Gecko-based browsers come with a reasonably good debugger. Gecko browsers use the Venkman debugger, while for Internet Explorer there are two versions: the Microsoft Script Debugger (free), and the Microsoft Script Editor (paid). Also, since the arrival of integrated toolbars and plugins, more and more support for JavaScript debugging is becoming available. The Gecko browsers have a native DOM inspector; for Internet Explorer a DOM inspector is integrated in the Web Developer toolbar.

Scripting languages are especially susceptible to bugs for the inexperienced programmer. Because JavaScript is interpreted, loosely-typed, and has varying environments (host applications), implementations and versions the programmer has to take exceptional care to make sure the code executes as expected.

Each script block is parsed separately. On pages where JavaScript in script blocks is mixed with HTML, syntax errors can be identified more readily by keeping discrete functions in separate script blocks, or (for preference), using many small linked .js files. This way, a syntax error will not cause parsing/compiling to fail for the whole page, and can enable a dignified die.

Related languages

There is no real relationship between Java and JavaScript; their similarities are mostly in syntax (that is, both derived from C). Their semantics are quite different: notably, their object models are unrelated and largely incompatible. Also worth mentioning is Microsoft's own VBScript, which, like JavaScript, is mainly used in web pages. VBScript has syntax derived from Visual Basic and is only available on Internet Explorer.

Due to the success of JavaScript, Microsoft developed a compatible language known as JScript. JScript was first supported in the Internet Explorer browser version 3.0 released in August, 1996. When web developers talk about using JavaScript in the IE browser, they usually mean JScript.

The need for common specifications for the two languages was the basis of the ECMA 262 standard for ECMAScript (see external links below), three editions of which have been published since the work started in November 1996 (and which in turn set the stage for the standardization of C# a few years later). Implementations of ECMAScript include JavaScript, JScript, and DMDScript. One term often related to JavaScript, the Document Object Model (DOM), is actually not part of the ECMAScript standard; it's a separate standard, developed by the W3C, closely related to XML.

ActionScript, the programming language used in Macromedia Flash, bears a resemblance to JavaScript. ActionScript has similar syntax to JavaScript, but the object model is dramatically different.

JSON, or JavaScript Object Notation, is a general-purpose data interchange format that is defined as a subset of JavaScript.

JavaScript OSA (JavaScript for OSA, or JSOSA), is a Macintosh scripting language based on the Mozilla 1.5 JavaScript implementation, SpiderMonkey. It is a freeware component made available by Late Night Software. Interaction with the operating system and with third-party applications is scripted via a *MacOS* object. Otherwise, the language is virtually identical to the core Mozilla implementation. It was offered as an alternative to the more commonly used AppleScript language.

Of only historical interest now, ECMAScript was included in the VRML97 standard for scripting nodes of VRML scene description files.

PROJECT DETAILS (ETrack)

I worked on project "**Etrack**" from 9 May 2005 to 20 May 2005. This was an in house project to manage resources and time effort of a project. This tool provides efficient management of project milestones, tasks and timesheets. It also provides facilities to HR department to manage attendance and leave the employees.

EVERY INDUSTRY IS FACING THE CONSTANT CHALLENGE OF DELIVERING EXCEPTIONAL CUSTOMER SERVICE WHILE MAINTAINING HEALTHY OPERATING MARGINS. WITH TODAY'S DIFFICULT ECONOMIC ENVIRONMENT, UNPRECEDENTED COMPETITION, RISING LABOR COSTS, AND RAPID EMPLOYEE TURNOVER, OVERCOMING THIS CHALLENGE HAS BECOMEMORE URGENT THAN EVER.ETRACK WORKFORCE MANAGEMENT SOLUTION FROM EBIX IS AN INTEGRATED SUITE OF WEBARCHITECTED APPLICATIONS DESIGNED SPECIFICALLY TO HELP CORPORATES PROVIDING EXCEPTIONAL SERVICE WHILE MANAGING ONE OF THEIR LARGEST CONTROLLABLE EXPENSES – LABOR. ETRACK WORKFORCE MANAGEMENT PROVIDES AUTOMATED BUSINESS FORECASTING, LABOR SCHEDULING, TIME AND ATTENDANCE, ENTERPRISE REPORTING, LEARNING MANAGEMENT AND WITH A LOAD OF OTHER FUNCTIONS – ENABLING CORPORATES TO REDUCE LABOR COSTS, IMPROVE SERVICE AND ACHIEVE REAL-TIME ENTERPRISE VISIBILITY AND CONTROL.

ETRACK WORKFORCE MANAGEMENT

REDUCE LABOR COSTS

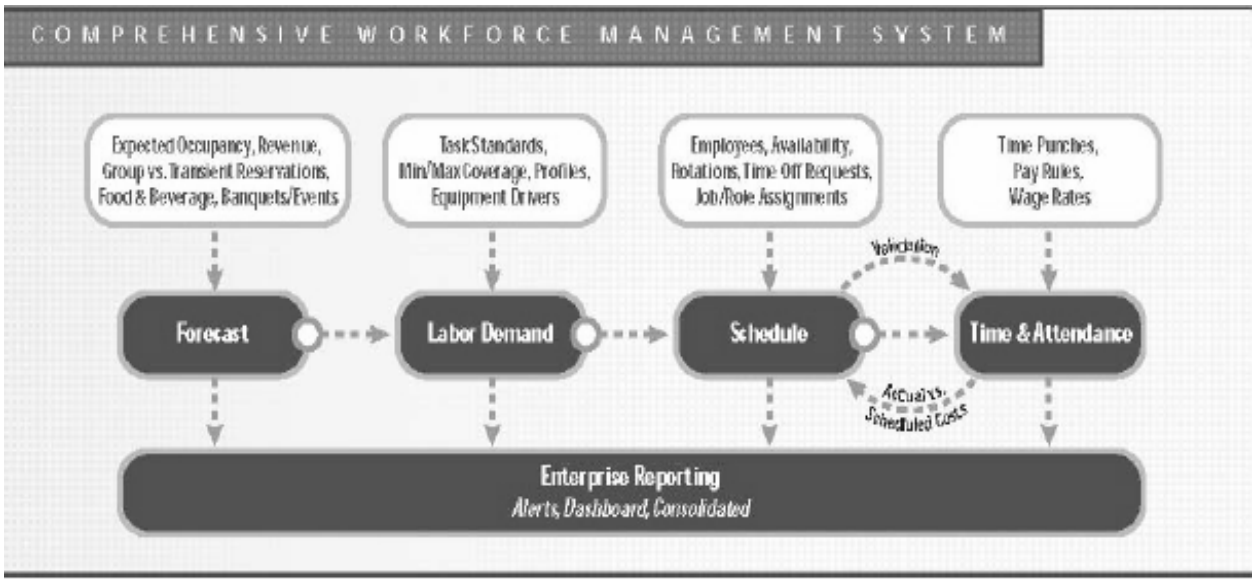
Reduce overtime and premium pay, eliminate costly payroll errors, and optimize staffing levels, share employees
across hotels, enforce adherence to assigned schedules, and apply pay rules accurately and consistently.

IMPROVE CUSTOMER RELATIONSHIP

With time sheet entry a track on employee's productiveness can be monitored. In the end results in better response on projects due to time management and maintaining a healthy customer relationship.

OPTIMIZE MANAGER TIME

Automate forecasting, scheduling, and time tracking processes to eliminate the time intensive manual tasks that keep managers inside their offices rather than attending to guest needs.



REDUCE EMPLOYEE TURNOVER

Improve workforce morale by accommodating employee schedule preferences, appraisals, internal news display; increase manager job satisfaction by reducing time-consuming administrative duties.

MANAGE PROACTIVELY, NOT REACTIVELY

Stop managing through a rear-view mirror. Real-time reporting and alerts put information in the hands of those that need it – your managers – in time to use it.

The ETrack Workforce Management solution is an integrated suite of Web architected applications designed to help cooperates manage labor across a complex operation. Capabilities include:

REPORTING

Generate improved business forecasts to project resource requirement. Detailed user reports can help manager to allocate responsibility Award report can give Managers an idea who should be the employee of the month. To have a grip on the ongoing project and critical activities Etrack helps the managers to have at glance reports on User skills

SCHEDULING

Optimize resource schedules based on inputs such as business forecasts, external regulations, internal policies, job assignments, and employee availability

Share employees' information across branches; accurately account for split costs across departments

TIME & ATTENDANCE

Record, track, and monitor labor hours and cost in real time Eliminate early and late punches through schedule validation at the system Alert managers of employees approaching overtime and guide them through the steps to adjust the schedule

LEARNING MANAGEMENT
Schedule, track, and measure results of employee training and automatically generate suggestions for additional training .Ensure employees scheduled for specific jobs have the required training and certifications

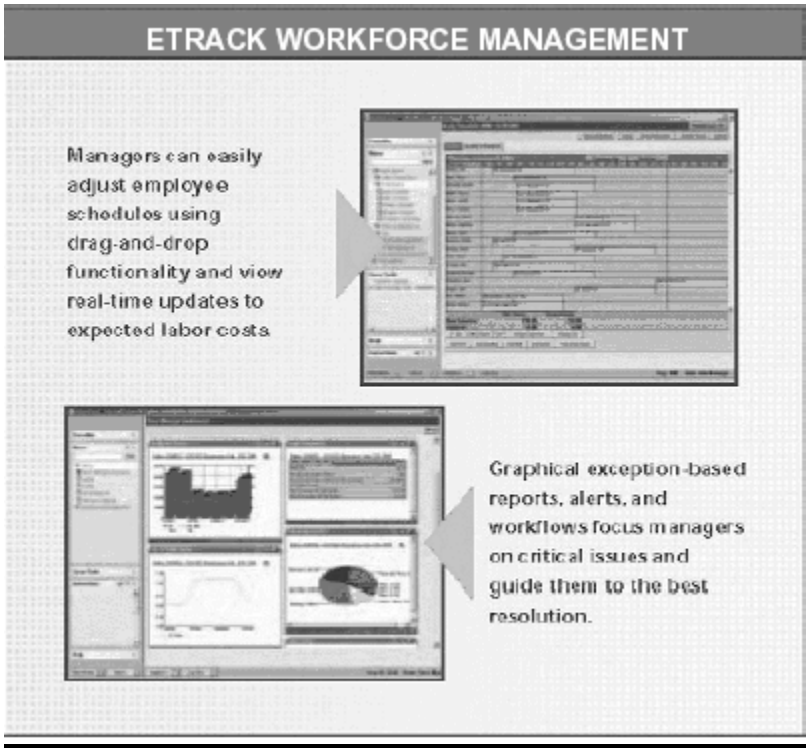
DASHBOARD
Provide real-time labor visibility across the enterprise with business summaries, alerts, and drill down reports .Alert managers of impending overtime, under-staffing, or other situations that affect project costs or SLA's

EMPLOYEE SELF SERVICE
View schedules, availability, personal profiles, and time cards online
Submit time-off requests online

TIME CLOCK
Capture employee login activity through over network
Ensure accurate time tracking, monitoring, and pay through seamless flow of employee data between the clock and time & attendance systems

INTERFACES
Integrate with third party Human Resources Management Systems, time clock devices, and payroll systems

Snapshots:-



Other details are as follows:-

Platform: - Microsoft Windows

Language: - C#

MY ROLE

I developed the whole attendance module of this project. Developing of this module consisted of:-

- Understanding of the company's holiday and working policies like number of paid leaves etc.
- Understanding of the already populated database.

This page consists of a big data grid where each user is given a row. The month and year can be selected from the drop down boxes. The visibility of all these depends on the user privileges. If a normal user is logged in he can view information regarding him and he cannot modify it, whereas if HR manager has logged in he can view details of all the employees and even modify it.

The server side was coded in C# and the client side was coded in JavaScript.

Few operations at client side were:-

- A check box was there which enabled or disable displaying of Sundays or Saturdays or holidays.
- Whether to show ex-employees or only those who are currently employed.
- The counting of paid leaves and displaying them in a textbox.

Few operations at server side were:-

- Populating the data grid.
- Storing the modifications done in data grid in database.
- Authenticating the user and displaying the data grid accordingly.