

PROJECT
TRACKING
SYSTEM

CONTENTS

1. ABSTRACT

2. INTRODUCTION

- Organization's Profile
- Project Profile

3. SYSTEM ANALYSIS

- Requirement Analysis
- Software Requirement Specification
- Feasibility Study
 - Economic Feasibility
 - Technical Feasibility
 - Behavioral Feasibility
 - Operational Feasibility
- Proposed System

4. OPERATING ENVIRONMENT

- Hardware & Software specifications

5. SYSTEM DESIGN

- Introduction
- Context Diagrams
- Data Flow Diagrams
- E-R Diagrams
- UML Diagrams
- Data Dictionary

6. SCREENS

7. DESIGN CONVENTIONS

- GUI Standards

8. REPORTS

9. SYSTEM ENVIRONMENT

10. SYSTEM TESTING

11. CONCLUSION

12. BIBLIOGRAPHY

ABSTRACT

The Project entitled “**Project Tracking System**” deals with the various levels of project development and will account for time used in analysis, design, programming, testing and verification etc.

It is well known fact that software companies undertake huge projects more than one at a time. Hence there is a profound need for the organizations to manage all the projects efficiently and ensure that projects cycle goes on smoothly and they are completed on time.

During the lifetime of a project, the organization has to commemorate all the activities of the project. This tool makes it easier for the organization to monitor the projects. It maintains records and tracks various parameters that influence software project development process and helps the management to take decisions at various stages of the project development.

The product will assist the organization in maintaining record of every project it undertakes. All the information relevant to the projects like size, time, effort and departments involved, etc is maintained by this tool.

This project deals with five modules – **Client, Business development officer (BDO), Developer, Human Resource Manager (HR), Project manager (PM)**.

Client deals with checking the status of the project by the client. Only the authenticated client login to the web page and checks in which phase the project is.

BDO deals with collecting the requirements from the client and add the client details and project details into the database and also views and updates the respective details.

Developer deals with filling the timesheets, updating and viewing the timesheets and also checking the project status.

HR deals with providing resources to the project manager by checking the employee details like skill and designation. HR can also view the status of the project.

PM deals with planning the project i.e. dividing the project into different tasks and assigning those tasks to the developers. PM also checks the timesheets of

developers, fills his own timesheet, updates and views the timesheet. PM can also check the status of the project.

INTRODUCTION

ORGANIZATION PROFILE

1976 was the year when **CMC Limited** commenced operations as **Computer Maintenance Corporation**. CMC took up the challenge to service all installations left by IBM, when IBM decided to wind up its operations in India in 1978, and took over the maintenance of over 800 IBM installations spread across the country. Later it also maintained computers supplied by scores of other foreign manufacturers.

While maintenance activities progressed apace, encouraged and enthused **CMC** spread its wings and turned its attention to other areas of computer support. Computer centers were set up in different cities and whole ranges of consultancy services were offered. Computer education and training programs were started. The R&D Center at Hyderabad undertook pioneering projects in the areas of contemporary technology. Software development in data communications and database management became one of the Corporation's fastest growing activities. India's first commercial computer network began taking shape.

A significant milestone was project **INTERACT**, a major United Nation funded project aimed at fostering self-reliance through design, development and systems engineering of real-time, on-line and image processing systems, dedicated to development-oriented applications. A number of professionals from Nigeria, Peru, Philippines, Yugoslavia and other developing countries worked with **CMC** on the project.

In view of the diversified business activities besides maintenance, **Computer Maintenance Corporation** became "**CMC Limited**" in 1984.

ABOUT THE COMPANY

CMC Limited is a leading IT solutions company and a subsidiary of **Tata Consultancy Services Limited (TCS Ltd)**, one of the world's leading information technology consulting, services and business process outsourcing organizations. It's a part of the **Tata group**, India's best-known business conglomerate.

Since its incorporation in 1975, CMC has an enviable record of successfully building IT solutions for massive and complex infrastructure and market projects. With 18 offices, 150 service locations, 520 non-resident locations and over 3,500 employees worldwide, it provides a wide spectrum of unique Information Technology solutions and services to a clientele of premier organizations in the government and private sectors.

CMC Americas, its subsidiary, services clients in the US, while its branch offices in the UK and Dubai market its products and services in Europe, Africa and the Middle East.

CMC achieve this with the help of its resource pool of engineers trained in diverse technologies, with vast domain knowledge and varied skill sets. Of its 3000 employees, about half have more than five years of experience. It also has a large competency pool that works on emerging technologies and competency areas. It have an impeccable educational profile as many as 44 per cent of its employees are qualified graduate engineers, while another 27 per cent have post-graduate or higher qualifications.

It also conducts significant research into emerging technologies and competence areas at its state-of-the-art, **ISO 9001-certified** R&D center in Hyderabad, India. That is what gives it an edge in complex, high technology projects

PROJECT PROFILE

ABOUT THE PROJECT

DEFINITION:

PROJECT TRACKING SYSTEM deals with the various levels of project development and will account for time used in phases viz analysis, design, coding, testing and implementation etc.

DESCRIPTION:

PROJECT TRACKING SYSTEM gives the management clear picture of usage of time by projects. By analyzing the results provided by the software they might rectify the defects in utilizing time and take remedial actions.

PROJECT TRACKING SYSTEM takes timesheet reports as input.

PROJECT TRACKING SYSTEM gives the individual report of the project, which contains time used for various tasks. In this tool CLIENT gives requirements to the BDO and makes an agreement with him. The new project information is entered by the BDO, based on the project information the PROJECT MANAGER will take resources from the HR and assign activities to DEVELOPERS who are working under him. Employees fill the time sheet and complete the task assigned to them. These completed tasks are tested and finally the project is submitted to the client.

This project deals with five modules – **Client, Bussiness development officer (BDO), Developer, Human Resource Manager (HR), Project manager (PM).**

Client deals with checking the status of the project by the client. Only the authenticated client login to the web page and checks in which phase the project is.

BDO deals with collecting the requirements from the client and add the client details and project details into the database and also views and updates the respective details.

Developer deals with filling the timesheets, updating and viewing the timesheets and also checking the project status.

HR deals with providing resources to the project manager by checking the employee details like skill and designation. HR can also view the status of the project.

PM deals with planning the project i.e. dividing the project into different tasks and assigning those tasks to the developers. PM also checks the timesheets of developers, fills his timesheet, updates and views the timesheet. PM can also check the status of the project.

Project Tracking System will do the following tasks:

- Record different phases of a project.
- Define phase wise tasks.
- Keep a track of project schedules.
- Record project definition.
- Phase wise project closures.
- Assign resources to a project.
- Access control for resources.
- Take daily efforts from each resource

OBJECTIVES:

The objective of this application is functionality and architecture of the **‘PROJECT TRACKING SYSTEM’**, and to estimate the effort of the project.

The general objectives are:

- Maintain project details.
- Maintain client details.
- Maintain employee details.
- Maintain timesheet for various levels of project.

SYSTEM ANALYSIS

REQUIREMENT ANALYSIS

A requirement is a feature that must be included in the actual design and implementation, getting to know the system to be implemented is of prime importance.

Main emphasis is on:

The inputs to the system.

The outputs expected from the system.

The people involved in the working of the system.

The volume of DATA (INPUTS) and the amount of INFORMATION (OUTPUTS) that will be involved.

With respect to the system itself, the following facts are taken into consideration:

The major processes involved.

The main points of application.

The processing rules for the collected data.

The exceptions that may be present.

The checks that should be placed in order to avoid wrong entries.

SOFTWARE REQUIREMENTS SPECIFICATION

Purpose: -

The purpose of this document is that it deals with various levels of project development and will account for time used in analysis, design, programming, testing and verification etc.

Scope: -

The tool involved in giving the management the clear picture of the usage of time by projects by analyzing the results provided by the software.

Documentation overview: -

This document has 3 major sections.

- **Section1** provides an overview of entire software requirement specification.
- **Section 2** provides the product that will be produced.
- **Section3** addresses the specific requirements of the system

FEASIBILITY STUDY

Feasibility study is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of the available resources. The objective of feasibility study is not to solve the problem but to acquire a sense of its scope.

Four key combinations are involved in the feasibility analysis. They are:

□ ECONOMICAL FEASIBILITY:

Economic analysis is the most frequently used method for evaluating the effectiveness of a client system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expecting from a client system and compare them with cost.

□ TECHNICAL FEASIBILITY:

Technical feasibility centers on the existing computer system (hardware, software etc) and to what extent it can support the proposed addition. The benefits such as high accuracy, minimum response time and user friendliness of the proposed system over weights cost for designing and implementing the new system.

□ BEHAVIOURAL FEASIBILITY:

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system. It is

common knowledge that computer installations have some thing to do with turnover transfers, retaining and changes in employee job status. Therefore, it is understandable that the introduction of a client system requires special effects to educate, sell and train the staff on new ways of conducting business.

□ **OPERATIONAL FEASIBILITY:**

The system is operationally feasible, it is made so easy that operator will not encounter any problem during working, as it is very user-friendly. Operational feasibility checks the scope of the system. The system under consideration should have enough operational research.

It is observed that the proposed system would provide a very interactive means to share information and have a far and wide range. The proposed system would make the information more interactive. Thus operation feasibility of the proposed system id found to be high.

PROPOSED SYSTEM

To take advantage of the latest technology and to facilitate clients to make online inquiries about their project status a tool needs to be developed. The tool should accomplish the following functions:

- The tool should be able to manage all the projects effectively and ensure that projects cycle goes on smoothly and they are completed on time.
- While accepting client's requirements, all necessary validations should be performed. It should also store the client's details.
- The system should generate the following reports:
 - ☐ Client's details list.
 - ☐ Employee details list.
 - ☐ Allocated resources list.
 - ☐ Project details list.
 - ☐ Allocated tasks list.

INVESTIGATION:

For the study of the project, I met the client and collected his requirements and got the information about various levels of project development and time assigned for each phase.

The main objective of this study is to determine whether the proposed system is feasible or not. Solution strategies are usually generated with out regard for feasibility because one cannot be both creative and critical at the same time. Hence feasibility analysis is a must to arrive at the most appropriate solution strategy. This feasibility analysis is done after the thorough study of the system.

The planning objective is achieved through a process of information discovery that leads to reasonable estimation.

Keeping the following aspects in mind, the system analysis for the development of the project is done.

Identify the drawbacks of the existing system.

Identify the need for conversion.

Perform feasibility study.

Identify hardware, software and the database requirements.

Create a system definition that performs the foundation for the subsequent work.

SOFTWARE SCOPE:

The first activity in software project planning is the determination of the software scope. The function and performance allocated to software should be assesses to establish a project scope. Software scope describes function, performance, reliability and constraints.

ESTIMATION:

Software project planning estimate can be done in many ways such as Lines Of Code (LOC) based estimation, Functional Point Estimation (FP), Process Based Estimation. We should select best of them depending on the requirement.

RESOURCE AVAILABILITY:

The resources, which are common to any system, consist of human effort, information and development resource. Developing any system with out satisfactory resources is in appropriate and impossible. The design development team for this project consists of five members or roles that are responsible for software development and internal testing and its usage. Information assembly is vital to this project. Various sources of information are available.

ROLES

Business Development Officer

He is a person who actually, interacts with the people who are demanding the project? He collects all the requirements of their projects.

Project Manager

The manager from the name itself indicates that to manage the projects. He plans the project and takes requirements from the HR. He then divides the project into tasks and assigns those tasks to the developers. He fills his own timesheet and also checks the developer's timesheets.

Developer

He deals with the actual coding part i.e. filling the time sheets and perform coding according to the time allotted to him. He then submits the filled timesheets and completed task to the system.

Human Resource Manager

His main activity is to provide resources to the project. He also checks the time sheets and decides the remuneration for the work done according to the time, in that time sheet.

Client

He is the person who demands the project and specifies the time, for which he needs the project back done. If the time is large slot then he has to accept the problems, and partial results. So that, he could place the new requirements for his project along with the advancement in the technology used.

OPERATING ENVIRONMENT

Software & Hardware Requirements

Hardware

For Server:

Minimum requirements include P III, 8 GB HDD,
128 MB RAM.

Recommended requirements include P IV, 12 GB HDD, 256 MB RAM.

For Client:

P III / P IV, 2GB HDD, 128 MB RAM.

Software

For Server:

ASP. Net, .NET Framework 2.0, SQL Server 2000 or higher.

For Client:

Windows 2000 professional / Win NT 4.0/ Windows XP,
. NET Framework 2.0

SYSTEM DESIGN

INTRODUCTION

Design is the first step in the development of any engineered product or system. It may be defined as the process of applying various techniques and principles for the purpose of defining a device, a processor or a system insufficient detail to physical realization.

In the design phase the detailed design of the system selected in the study takes place. System design is stated by reviewing the study phase activities and making decisions as to which functions are to be performed by hardware, software or manually. This process through which the initial requirements are specified in the study phase are translated into a representation of the software. Initially, the representation depicts holistic view of the software. Subsequent refinement leads to

a design representation, which is very close to that of the source code. The design phase recommendations are presented as reports to the user.

The following steps are followed in designing software:

The statements of functions desired from software to be developed are recognized.

The database design steps are applied to identify the various entities and relationships between them.

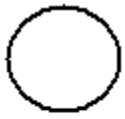
General access methods are then applied to the conceptual schema to get the logical model, which is then further transformed into relational tables.

The concept of s/w engineering is then applied to get the logical architecture of the system comprising of different modules each containing all the related functions.

CONTEXT DIAGRAM

The context diagram is a data flow diagram showing data flows between a generalized application within the domain and the other entities & abstractions with which it communicates. One thing that differentiates the use of data flow diagrams in domain analysis from other typical users is that the variability of the data flows across the domain boundary must be accounted for with either a set of diagrams or text describing the differences.

The symbols that we use are:



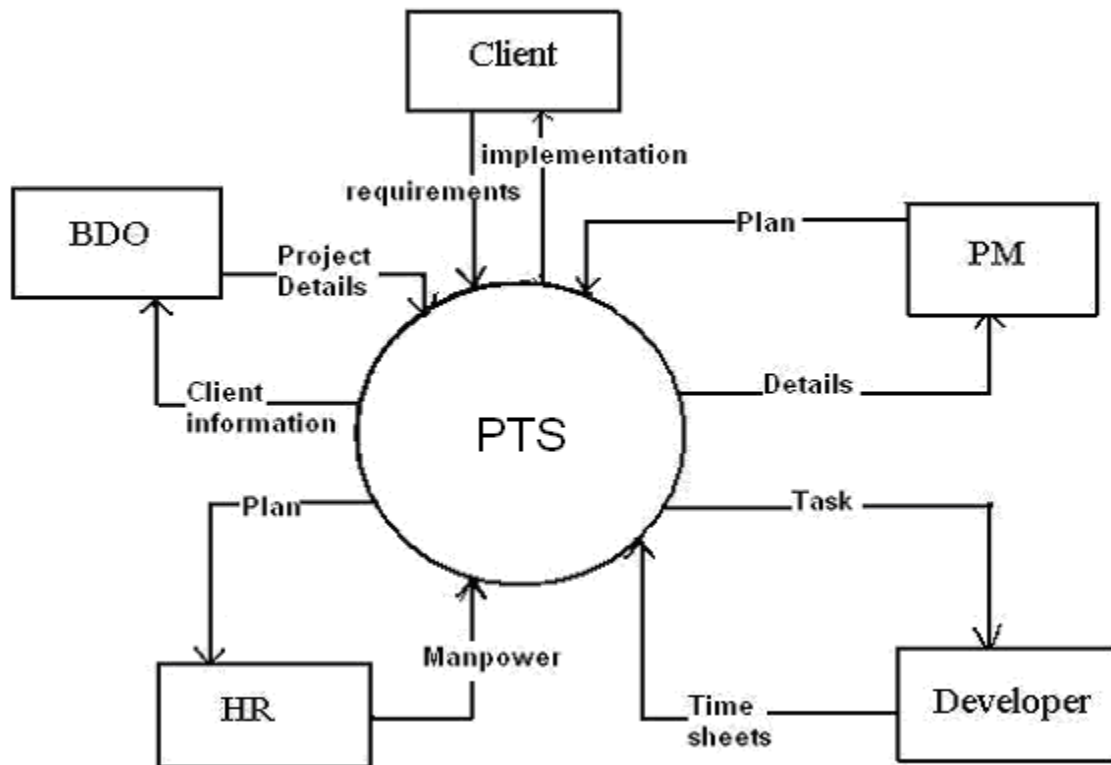
System



Entity



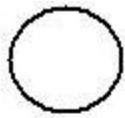
Data flow



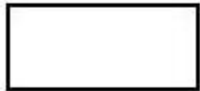
Data Flow Diagrams

DFD's represents the flow of data.

The symbols that we use are:



Process



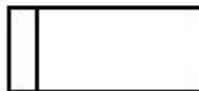
Entity



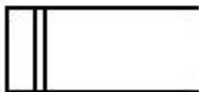
Data flow



Duplicated Entity

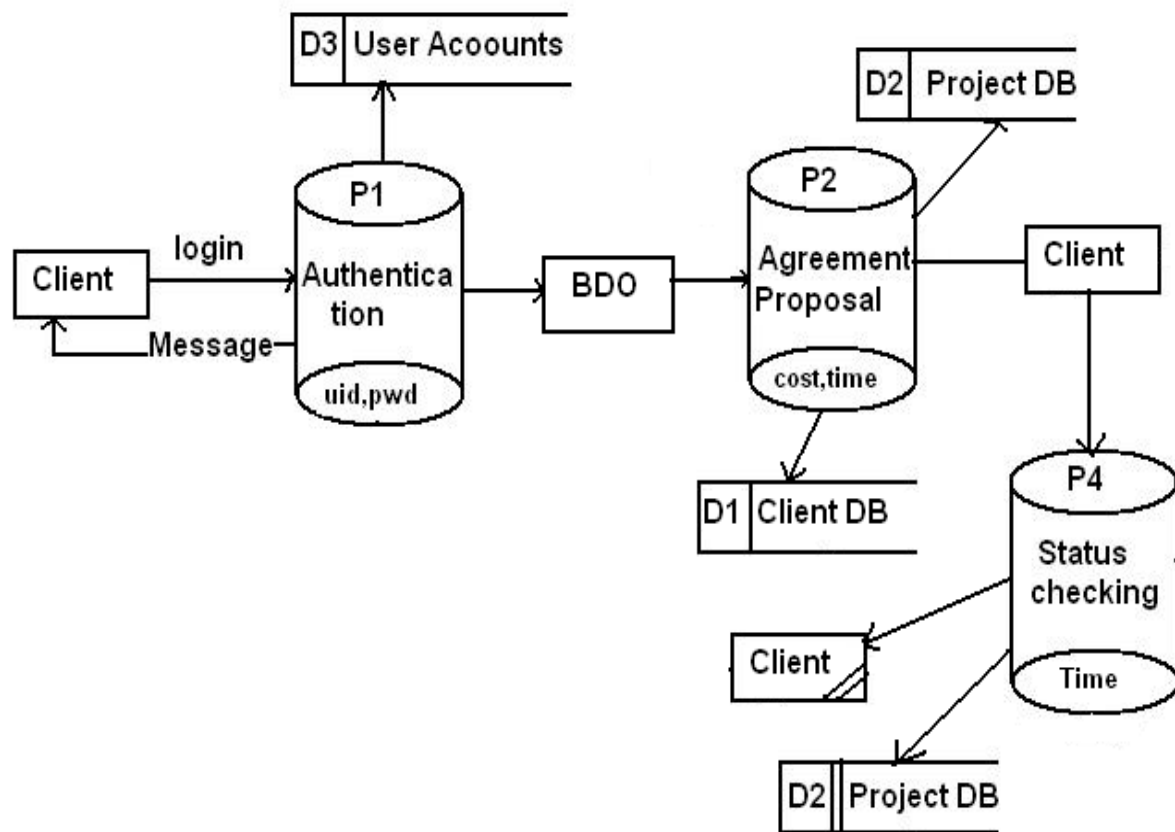


Database

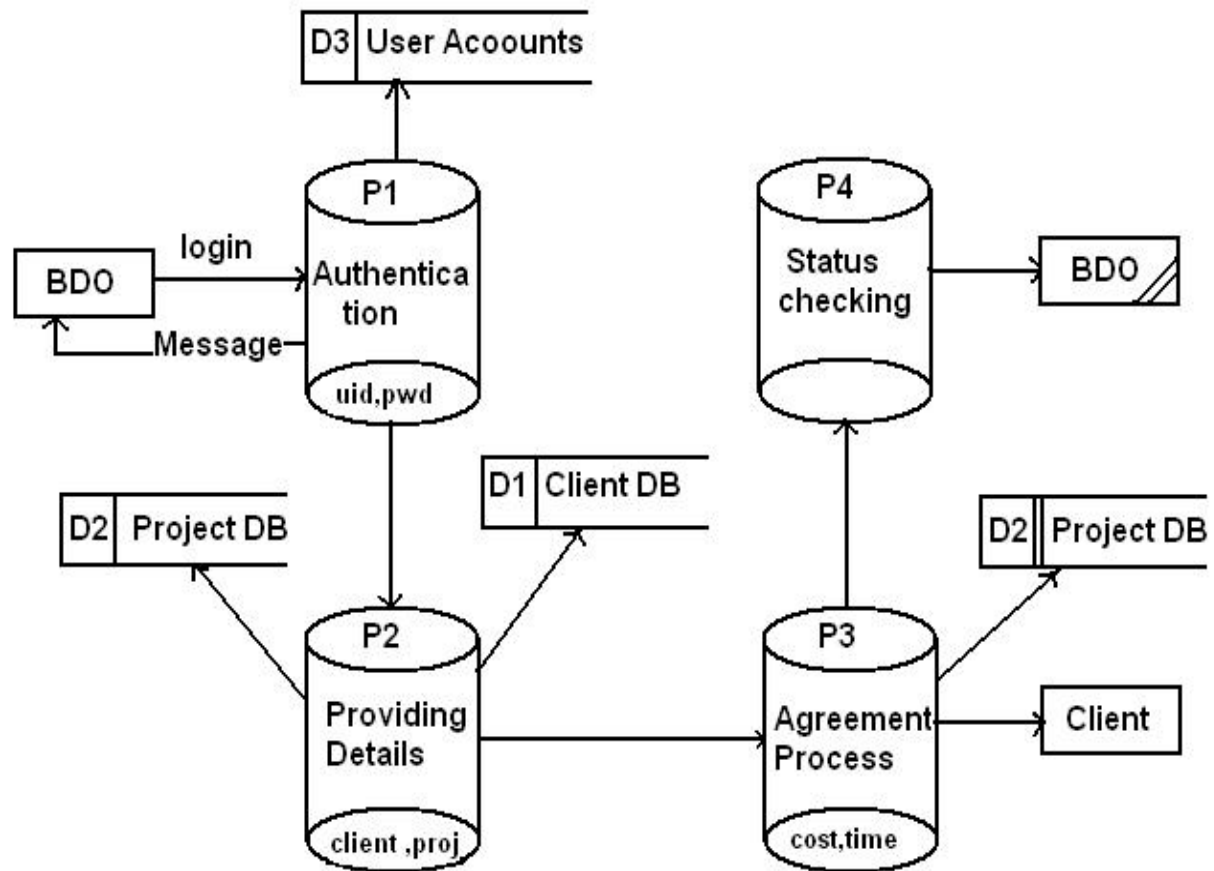


Duplicated Database

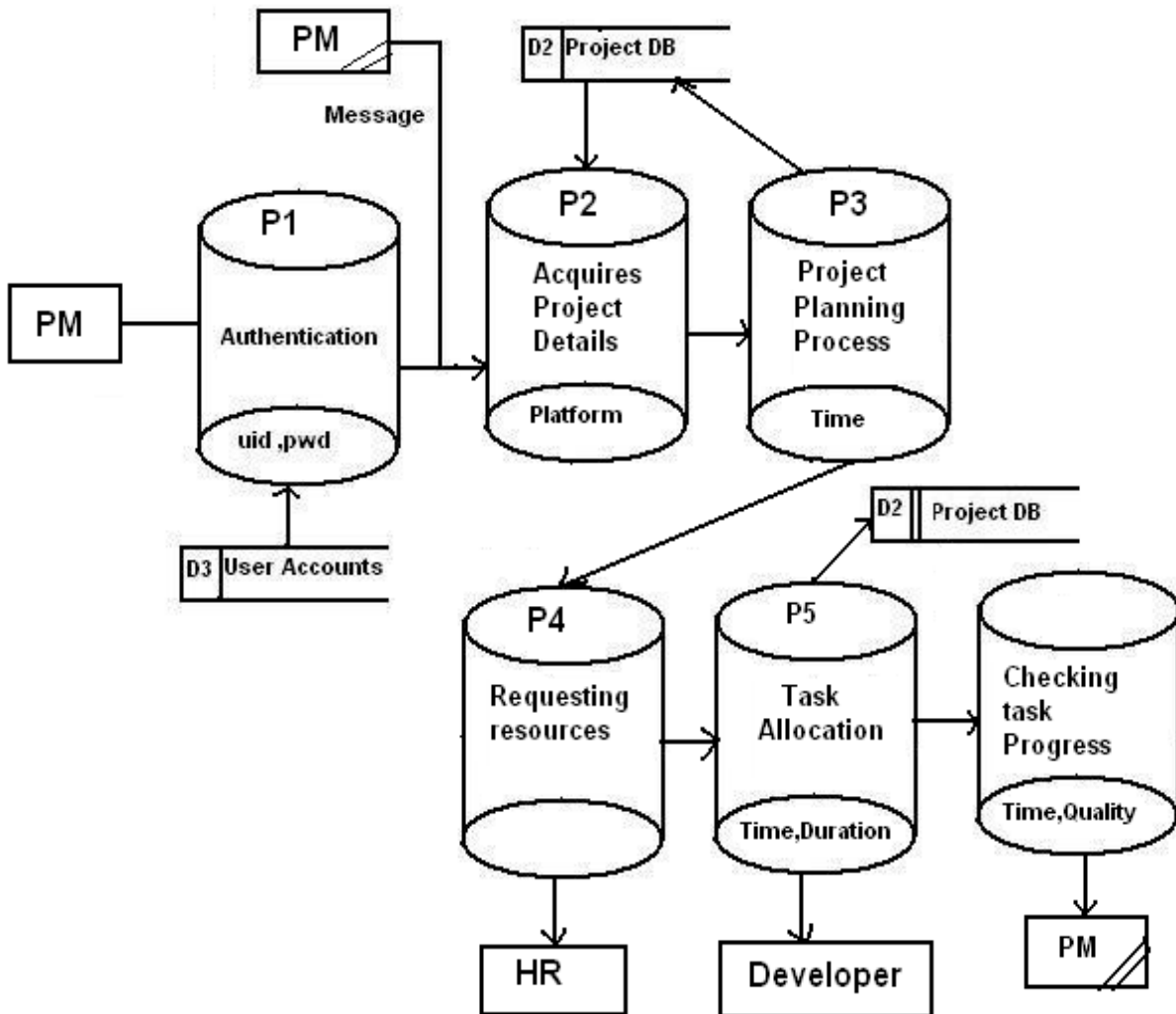
Client DFD



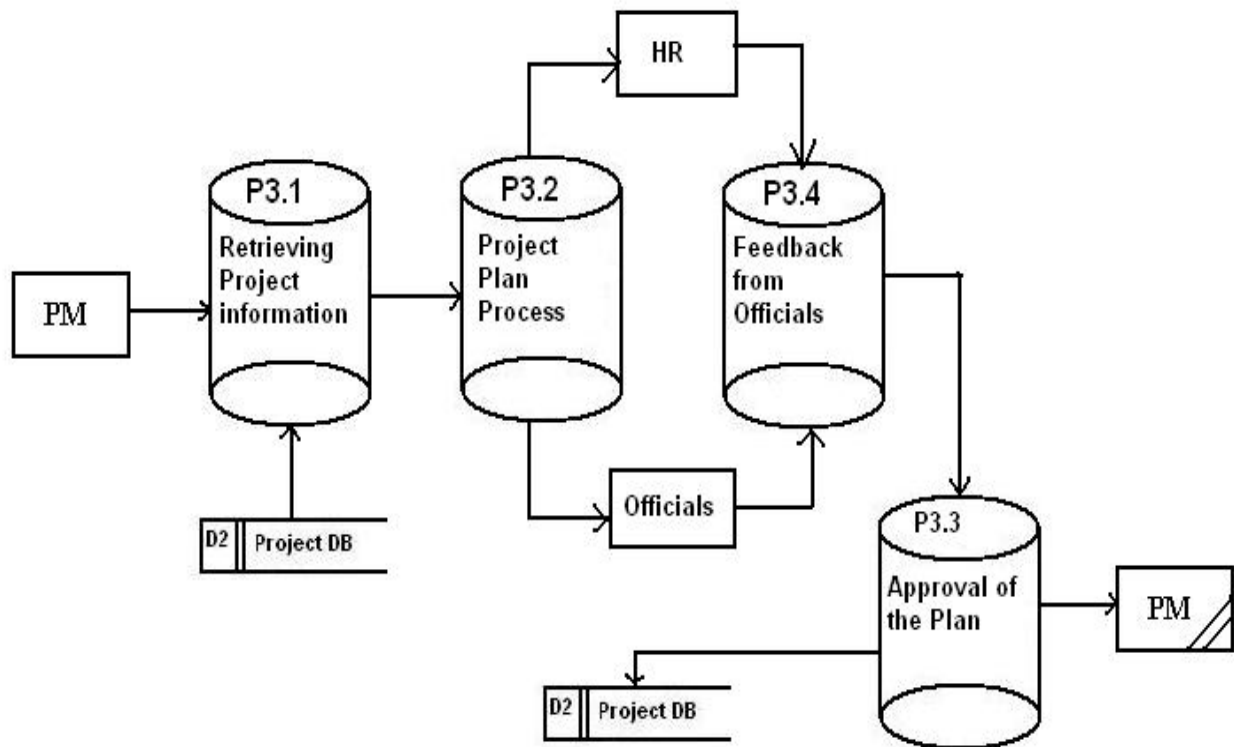
BDO DFD



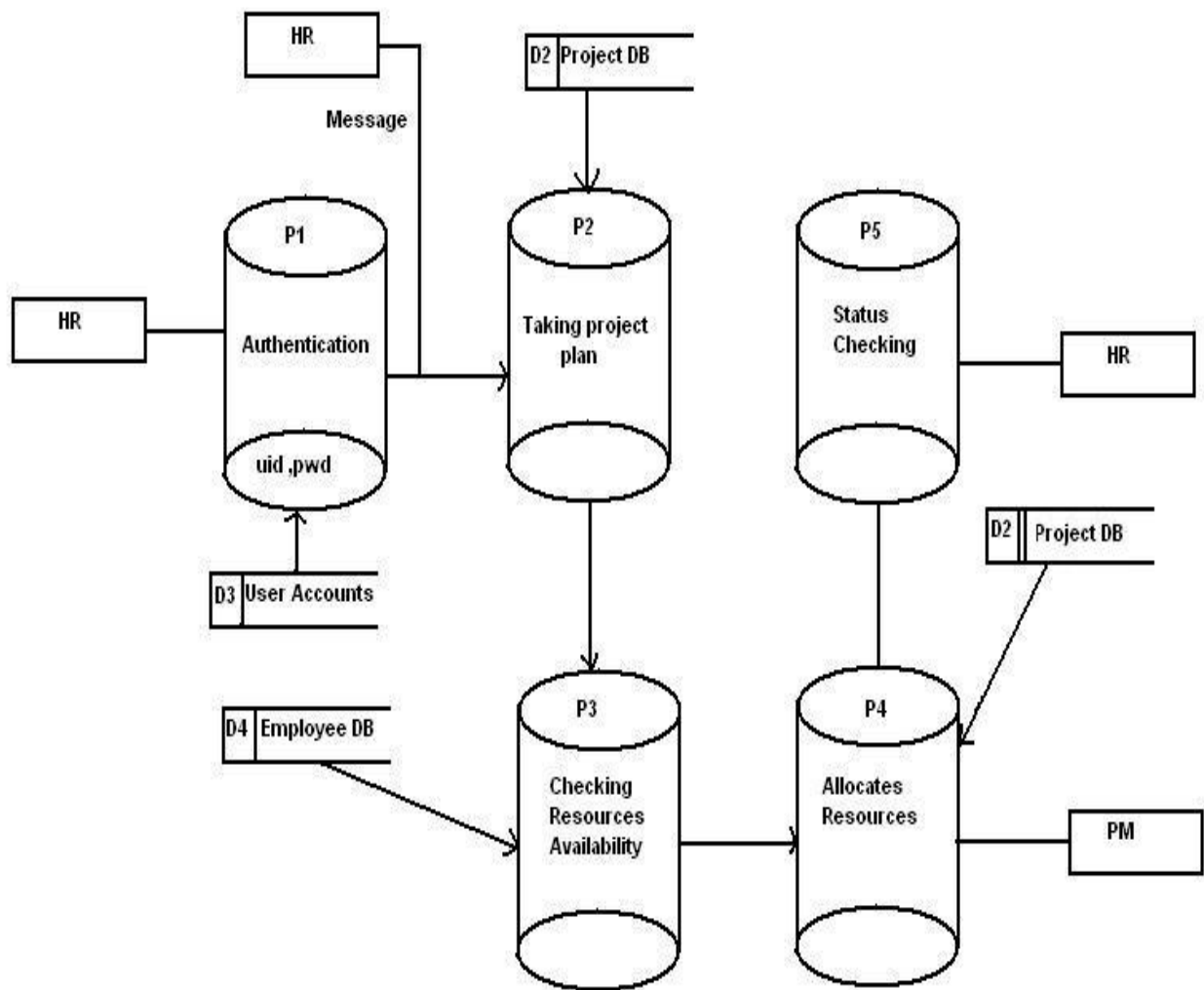
PM Level1 DFD



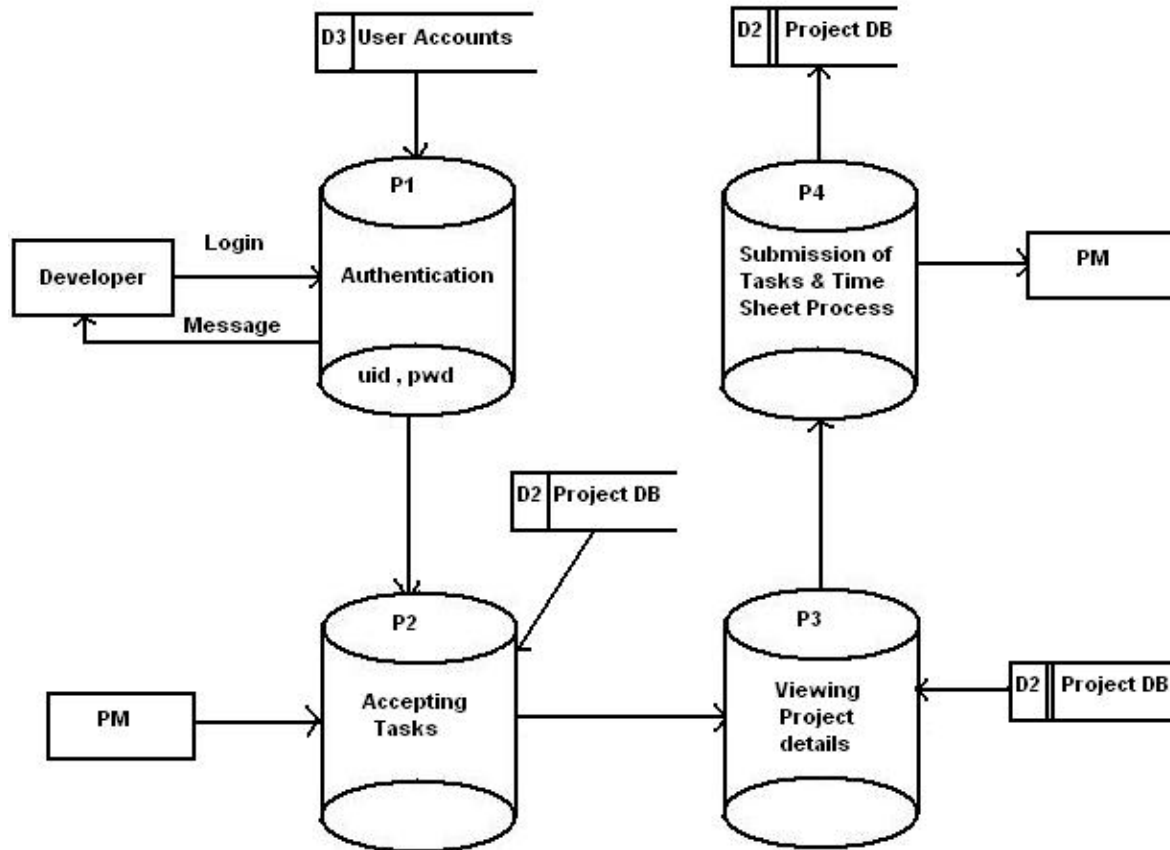
PM Level2 DFD



Human resource DFD

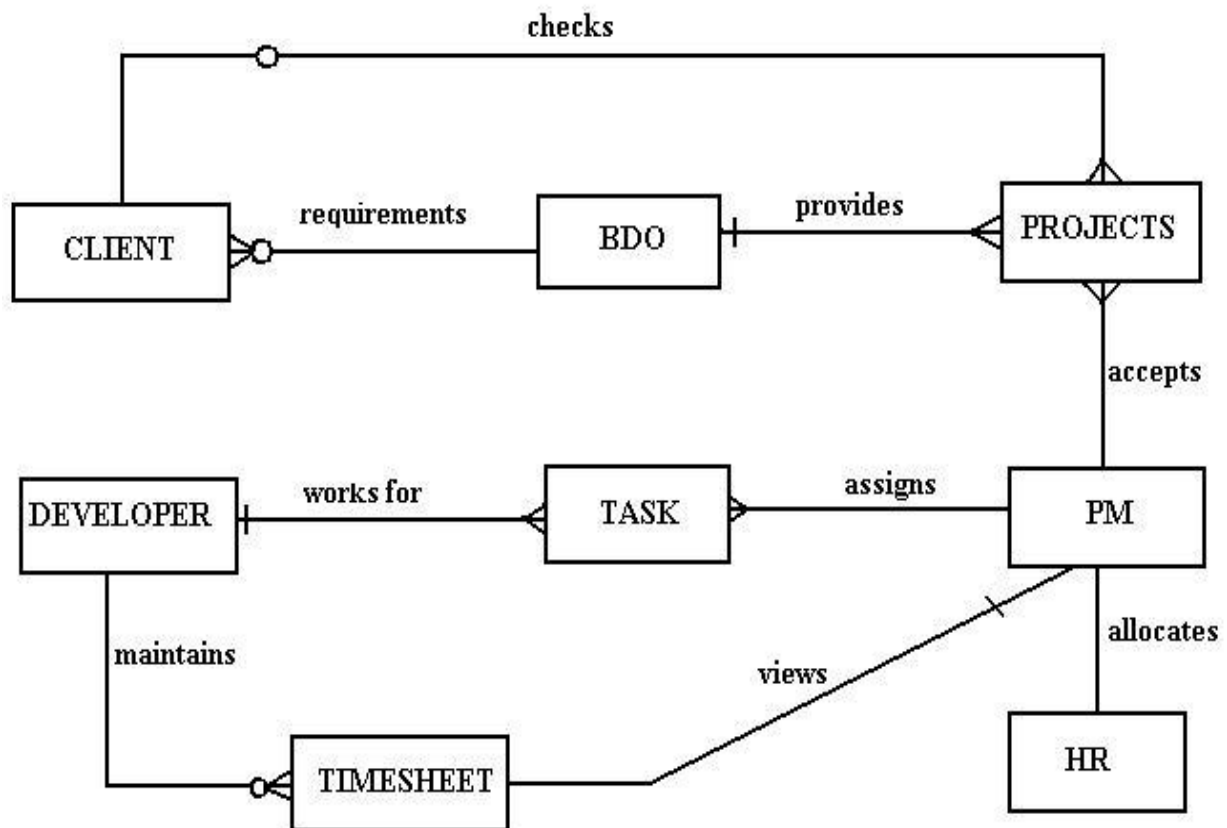


Developer DFD



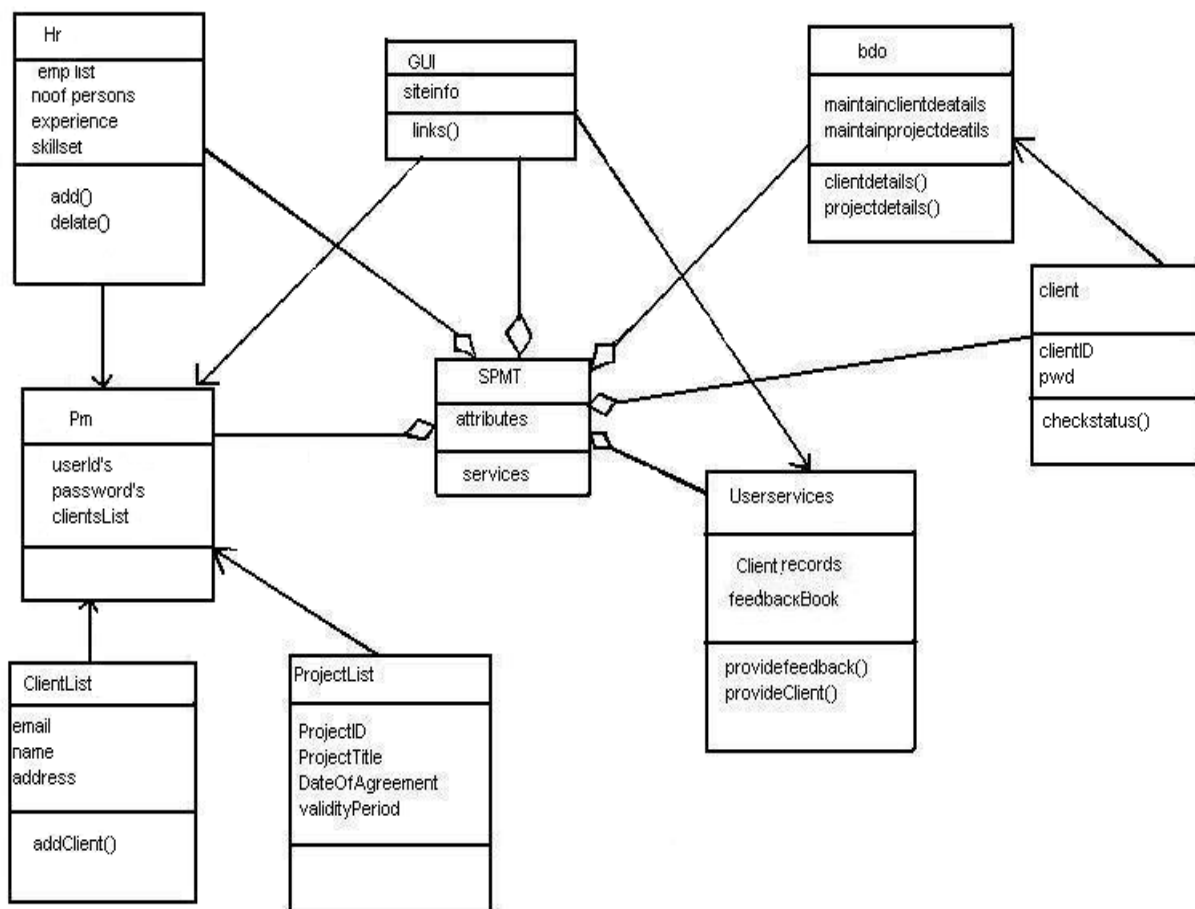
E-R DIAGRAM

The following diagram depicts various entities involved in the Project Tracking System and relationships between those entities.

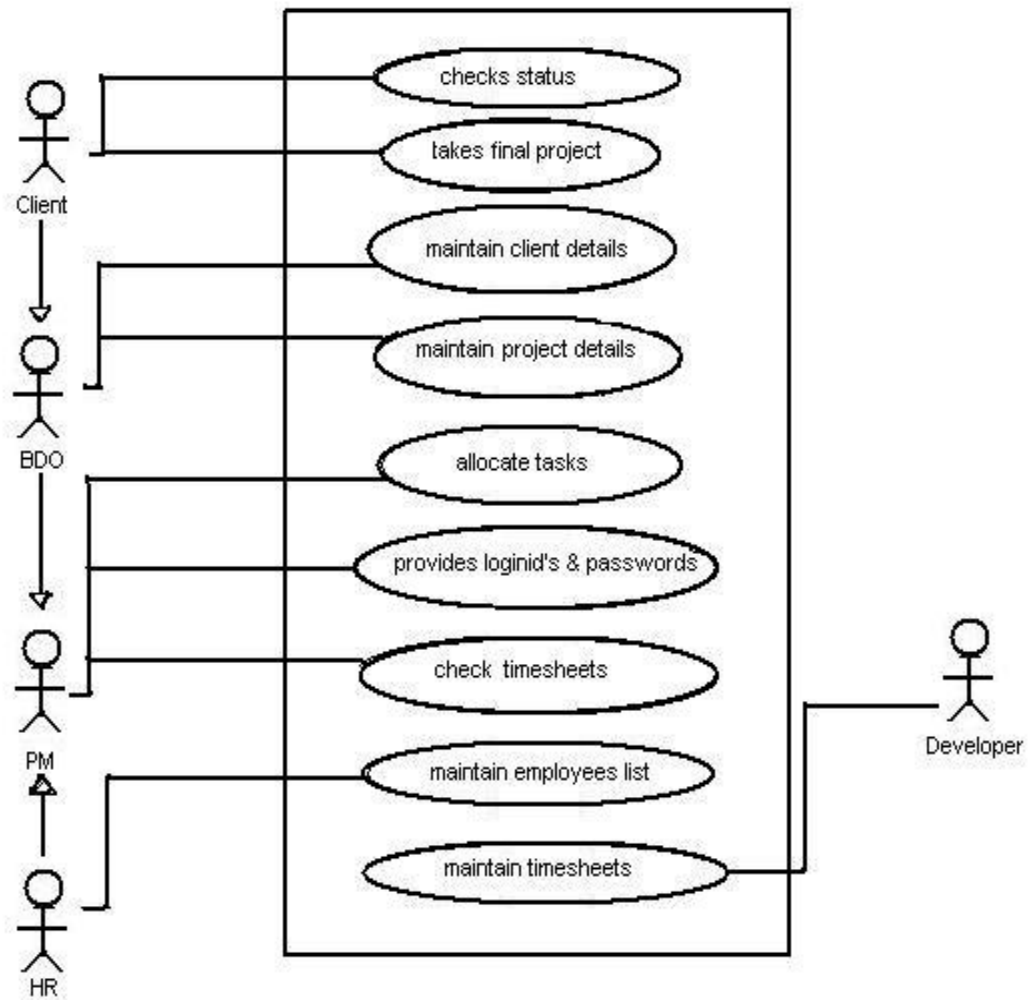


UML DIAGRAMS

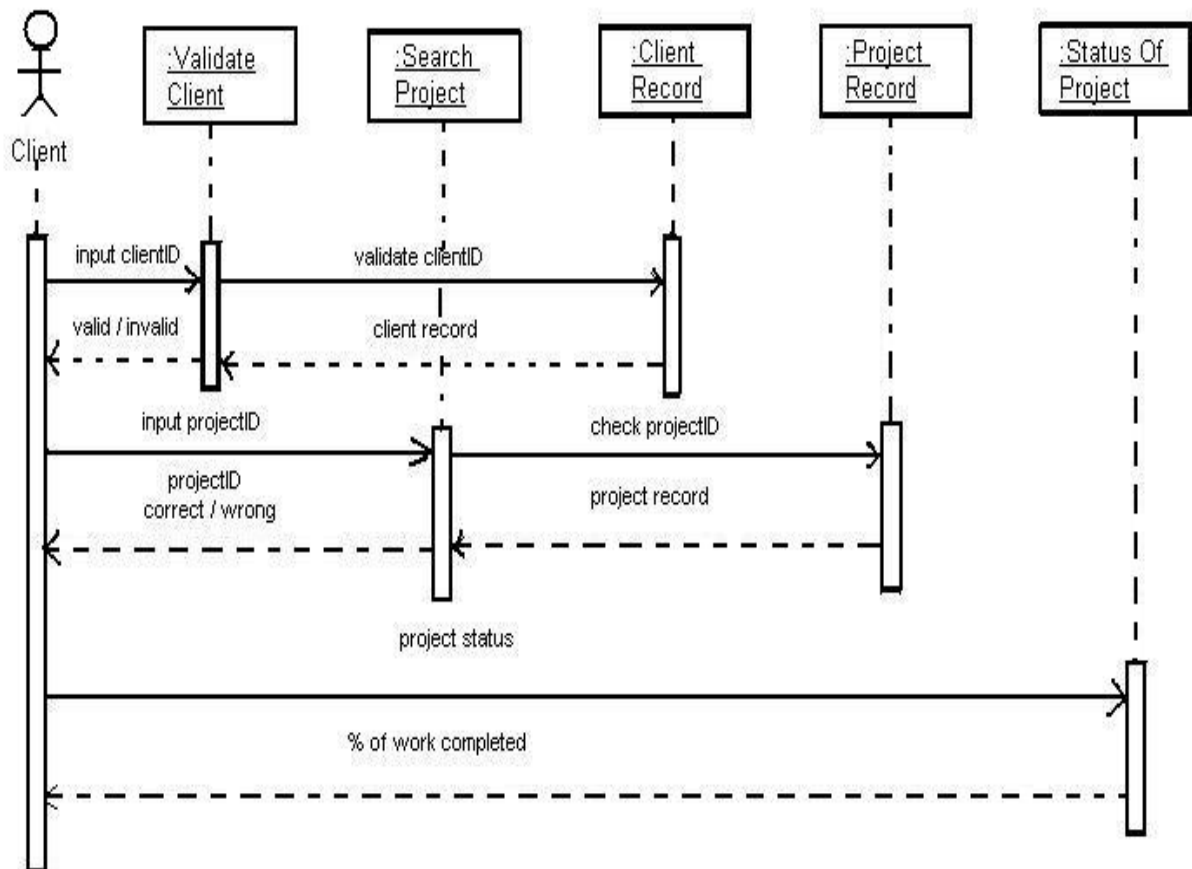
Class Diagram



Use Case Diagram



Client Sequence Diagram



DATA DICTIONARY

Data Dictionary Format

Key Type	Column Name	Column Description	Data Type	Length	Default values	Allow nulls	Domains	Input Format

CLIENT

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
CID	VARCHAR2 (20)	Primary Key	
CNAME	VARCHAR2 (40)		
PNO	NUMBER (30)		
EMAIL	VARCHAR2 (50)		
POSITION	VARCHAR2 (50)		
OFFADD	VARCHAR2 (80)		
HOMEADD	VARCHAR2 (80)		
TYPEOFIND	VARCHAR2 (50)		
SCALE	VARCHAR2 (20)		
PERIOD	NUMBER (20)		

EMPLOYEE

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
EID	VARCHAR2 (50)	PRIMARYKEY	
ENAME	VARCHAR2 (50)		
SKILLSET	VARCHAR2 (50)		
DESIGNATION	VARCHAR2(50)		
EXPERIENCE	VARCHAR2(50)		
CTC	NUMBER(20)		

PHNO	VARCHAR2(50)		
EMAIL	VARCHAR2(50)		

HR REQUIREMENTS

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
PID	VARCHAR2 (30)		
SKILL	VARCHAR2 (30)		
NOP REQ	NUMBER (3)		
SDATE	VARCHAR2 (30)		
EDATE	VARCHAR2 (30)		
REQ_ID	VARCHAR2 (30)		

MODULE

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
MID	VARCHAR2 (30)		
MNAME	VARCHAR2 (30)		
NOP	NUMBER (3)		
NOTASKS	NUMBER (3)		
PID	VARCHAR2 (50)		
MDETAILS	VARCHAR2 (80)		

MODULE STATUS

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
DATE1	VARCHAR2 (20)		
TIME	VARCHAR2 (20)		
MID	VARCHAR2 (30)		
STATUS	VARCHAR2 (500)		

PROJECT

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
PID	VARCHAR2 (50)	Primary key	

NOP	NUMBER (5)		
PTITLE	VARCHAR2 (50)		
CID	VARCHAR2 (50)		
SDATE	VARCHAR2 (50)		
EDATE	VARCHAR2 (50)		
COST	NUMBER (20)		
ESYSTEM	VARCHAR2 (50)		
PSYSTEM	VARCHAR2 (50)		
ABSTRACT	VARCHAR2(50)		
LOC	VARCHAR2(50)		
TYPEOF OS	VARCHAR2(50)		
TYPEOF APPL	VARCHAR2(50)		
PLATFORM	VARCHAR2(50)		
SW_MODEL	VARCHAR2(50)		
PM	VARCHAR2(50)		

PROJECT ASSIGN

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
EID	VARCHAR2(30)		
PID	VARCHAR2 (30)		
REPORT DATE	VARCHAR2 (30)		
REPORT MGR	VARCHAR2(30)		
NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
DATE1	VARCHAR2(30)		
TIME	VARCHAR2(30)		
PID	VARCHAR2(30)		
STATUS	VARCHAR2(500)		

TASK

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
TID	VARCHAR2(30)		
TNAME	VARCHAR2 (30)		
MID	VARCHAR2 (30)		
NOP	NUMBER (3)		
SDATE	VARCHAR2 (30)		

EDATE	VARCHAR2 (30)		
T PARTICULARS	VARCHAR2 (80)		
DEPENDABILITY	VARCHAR2 (50)		
STIME	VARCHAR2 (30)		

TIME SHEET

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
TID	VARCHAR2 (30)		
STATUS	VARCHAR2 (500)		
S TIME	VARCHAR2 (30)		
E TIME	VARCHAR 2(30)		
S DATE	VARCHAR2 (30)		
E DATE	VARCHAR 2(30)		
CURR DATE	VARCHAR2 (30)		

TASK ASSIGN

NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
TID	VARCHAR2 (50)		
EID	VARCHAR2 (50)		

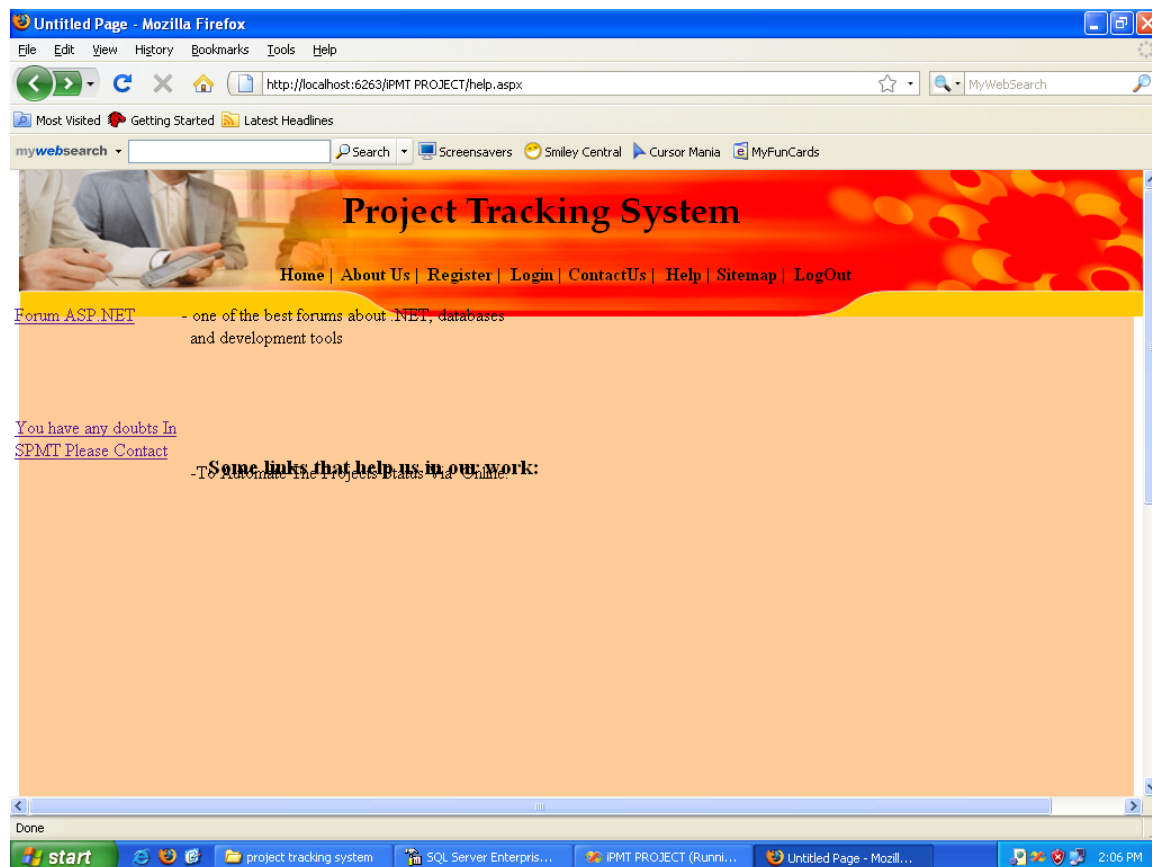
TASK STATUS

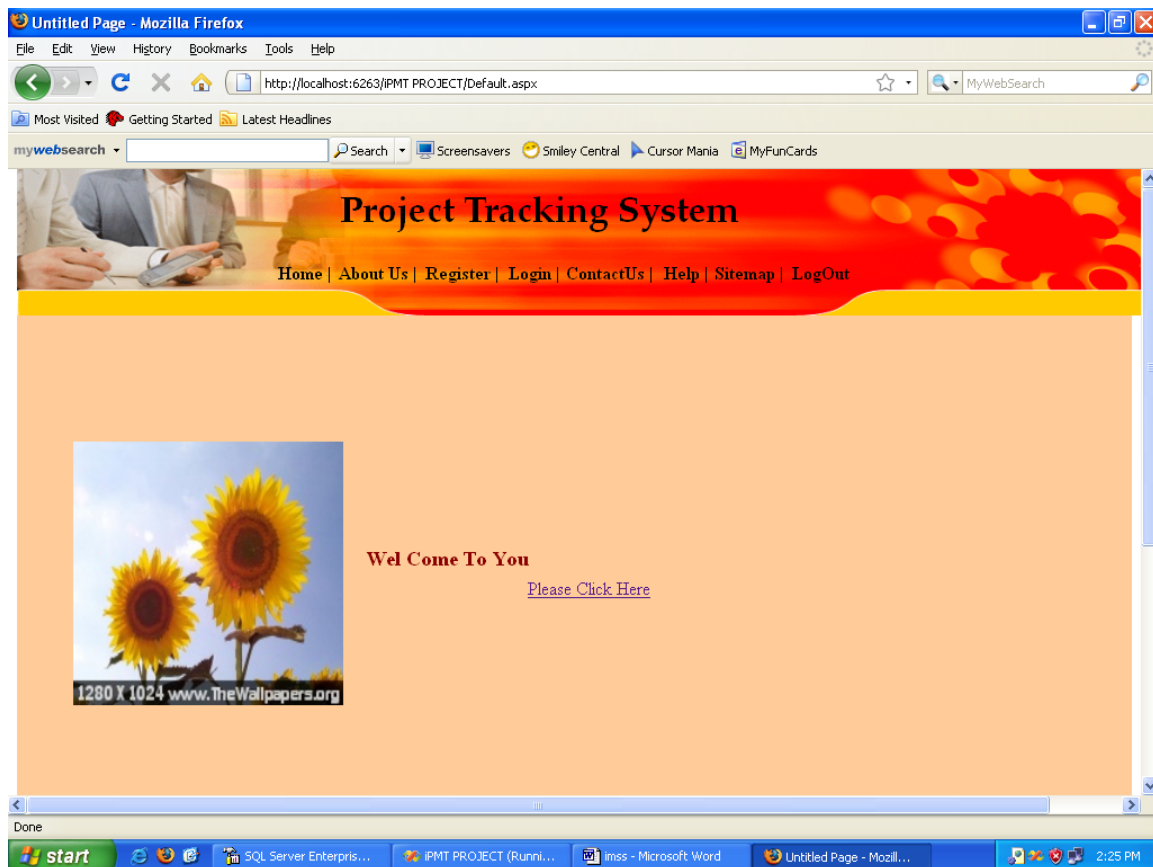
NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
TID	VARCHAR2 (50)		
DATE	VARCHAR2 (50)		
TIME	VARCHAR2 (50)		
STATUS	VARCHAR2 (50)		

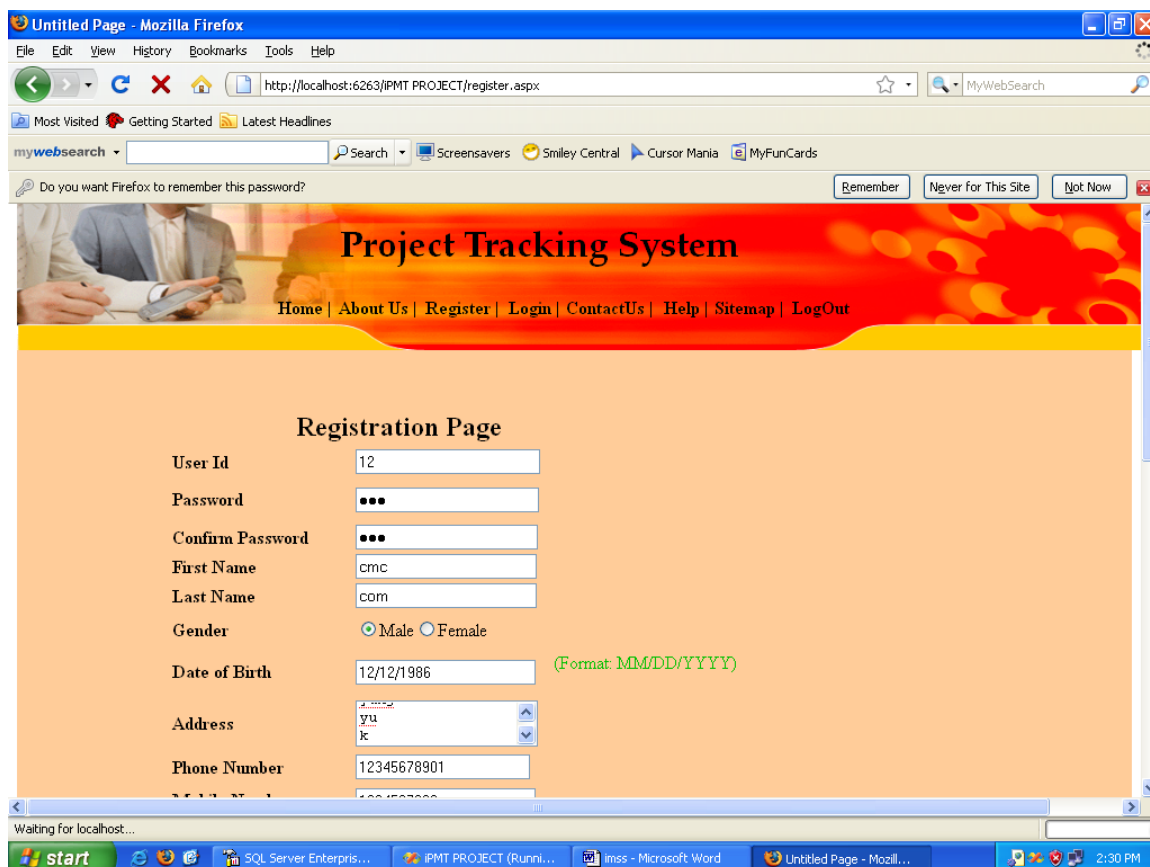
USER PASSWORD

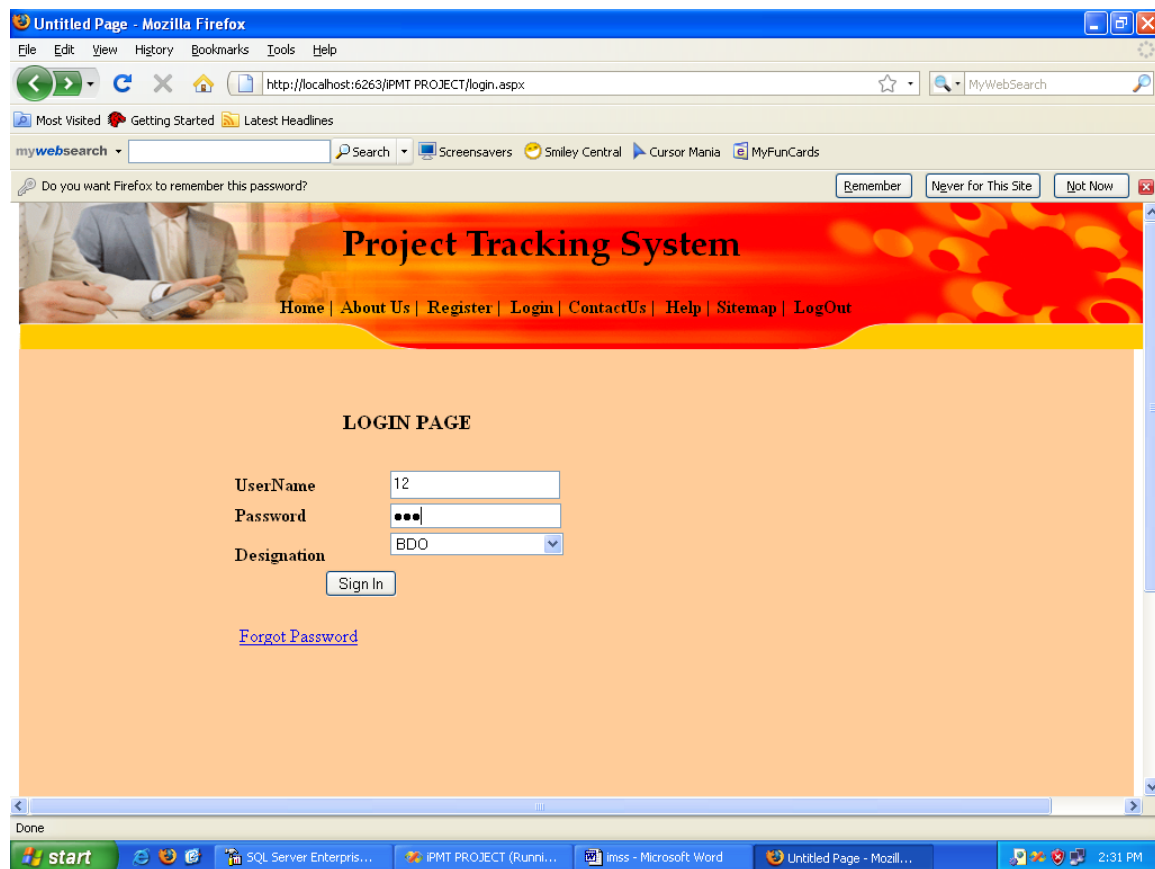
NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
UNAME	VARCHAR2 (10)		
PASSWORD	VARCHAR2 (50)		

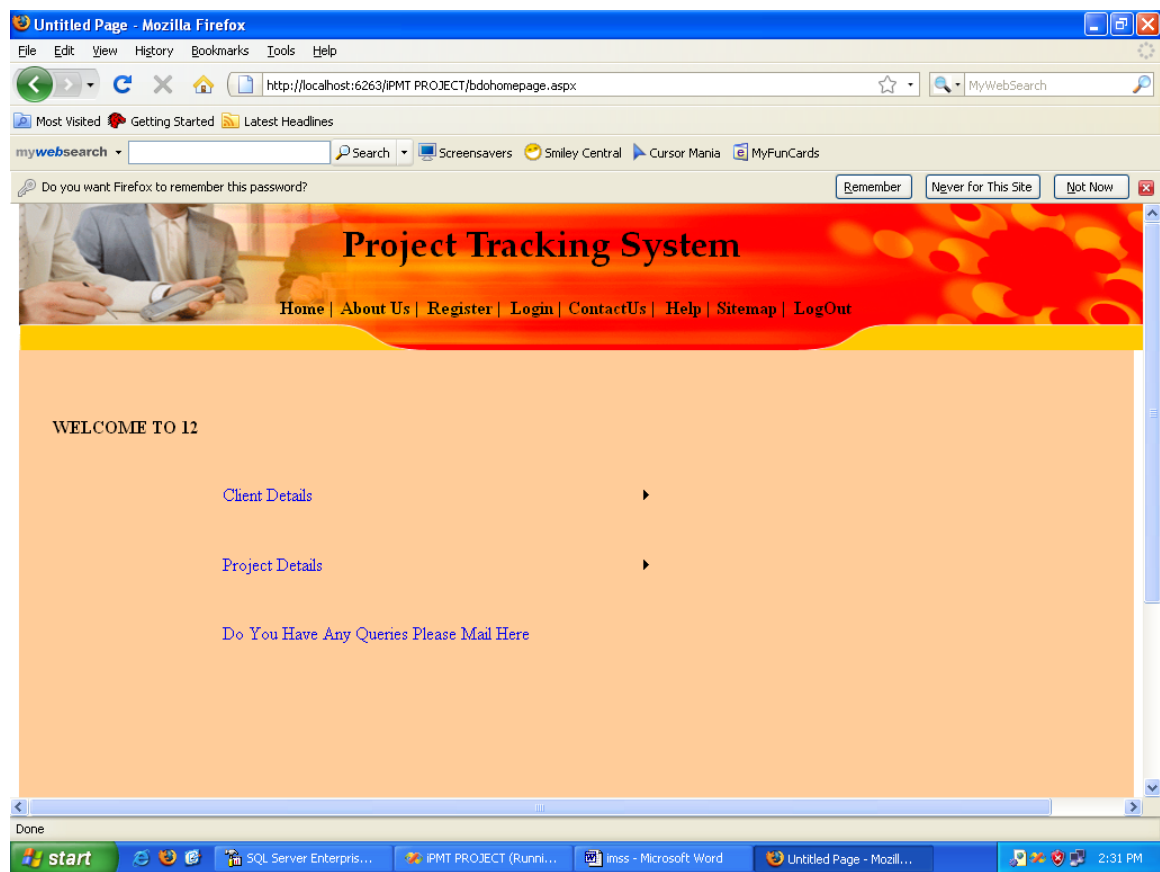
SCREENS

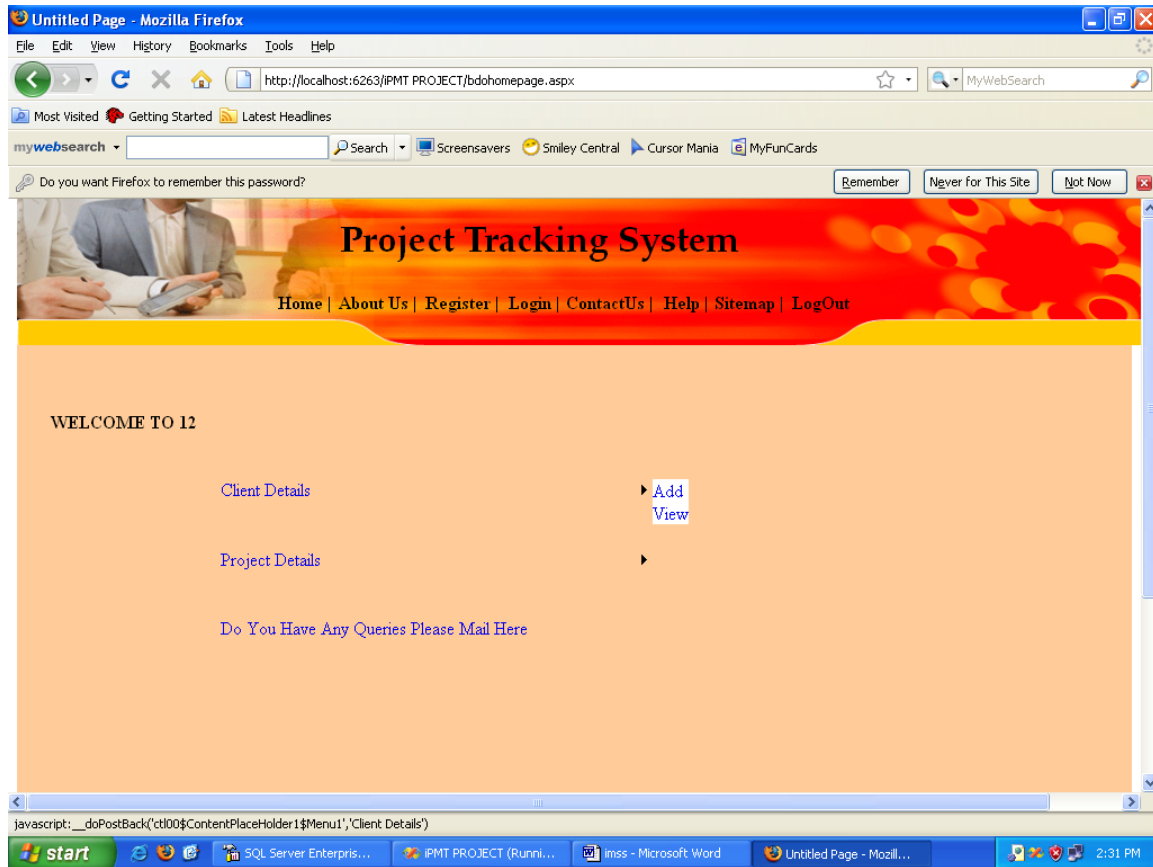


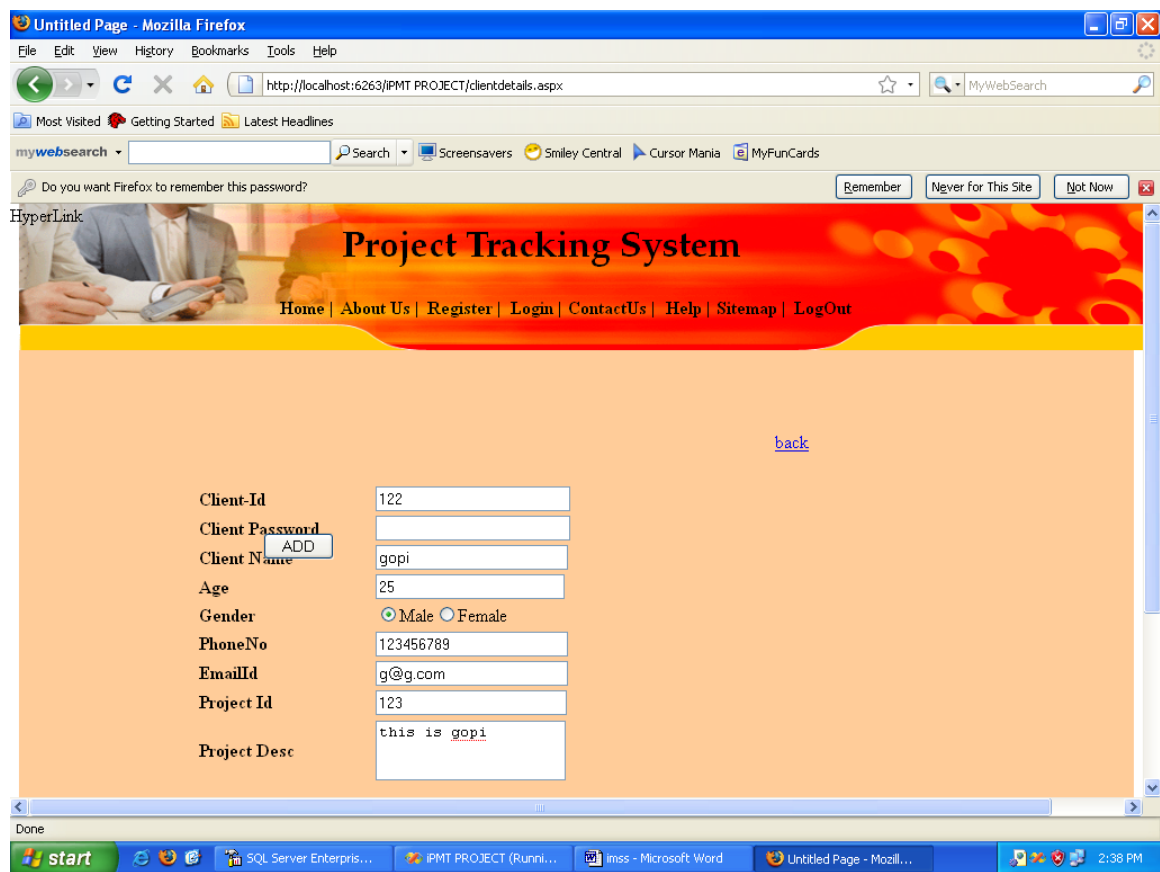


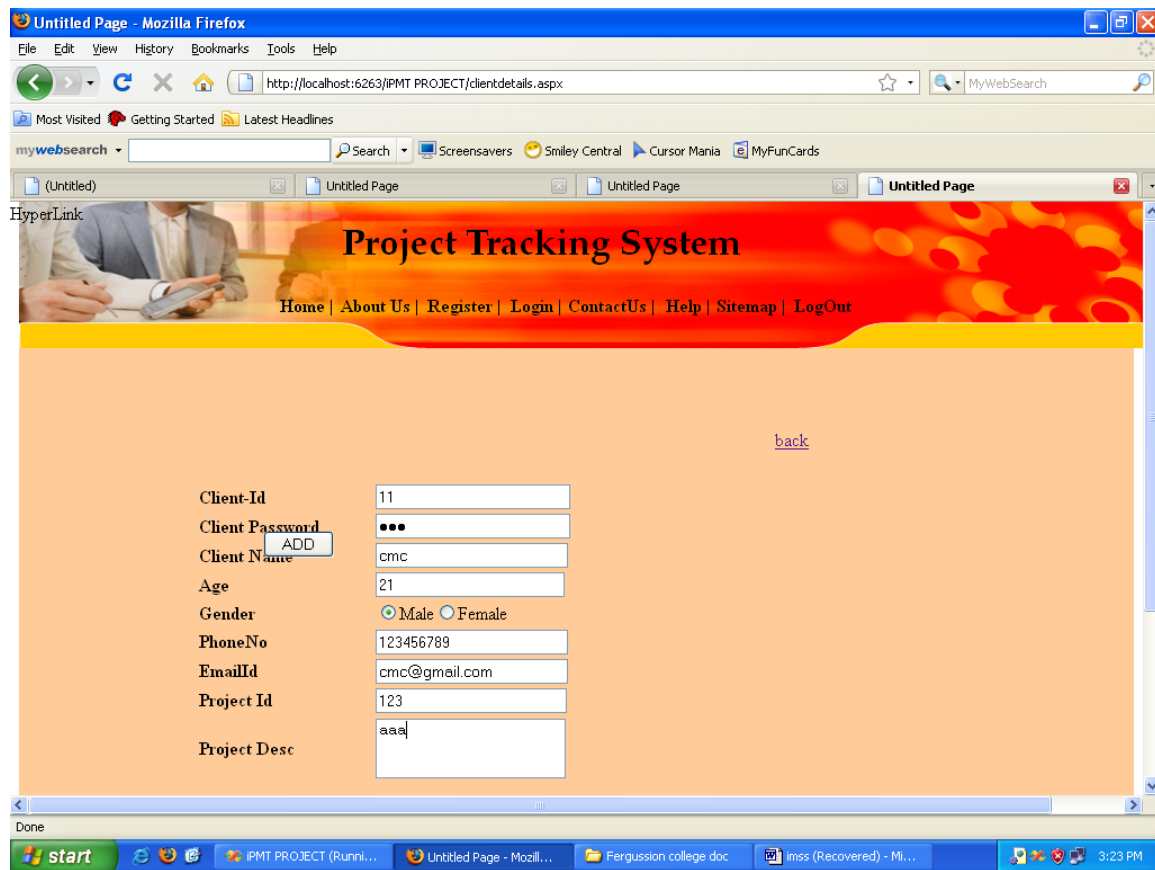












Untitled Page - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:6263/IPMT PROJECT/viewproject.aspx

MyWebSearch

Most Visited Getting Started Latest Headlines

mywebsearch Search Screensavers Smiley Central Cursor Mania MyFunCards

Untitled Page



Project Tracking System

Home | About Us | Register | Login | ContactUs | Help | Sitemap | LogOut

[back](#)

projectid	projectname	clientid	projectdesc	projectstartdate	projectenddate	projectstatus	teamid	teamsize	moduleid	modulename
p420	Smpt	a.g.khan	FHUIASDHGKBJ	3/22/2008 12:00:00 AM	4/22/2008 12:00:00 AM	Started	T420	4	m420	Maintaing employee details
p27	WareHouseManagement	praneeth	kdhsfhmk	3/22/2008 12:00:00 AM	4/22/2008 12:00:00 AM	started	TT27	3	M27	project maintaining

1 2

Done

start

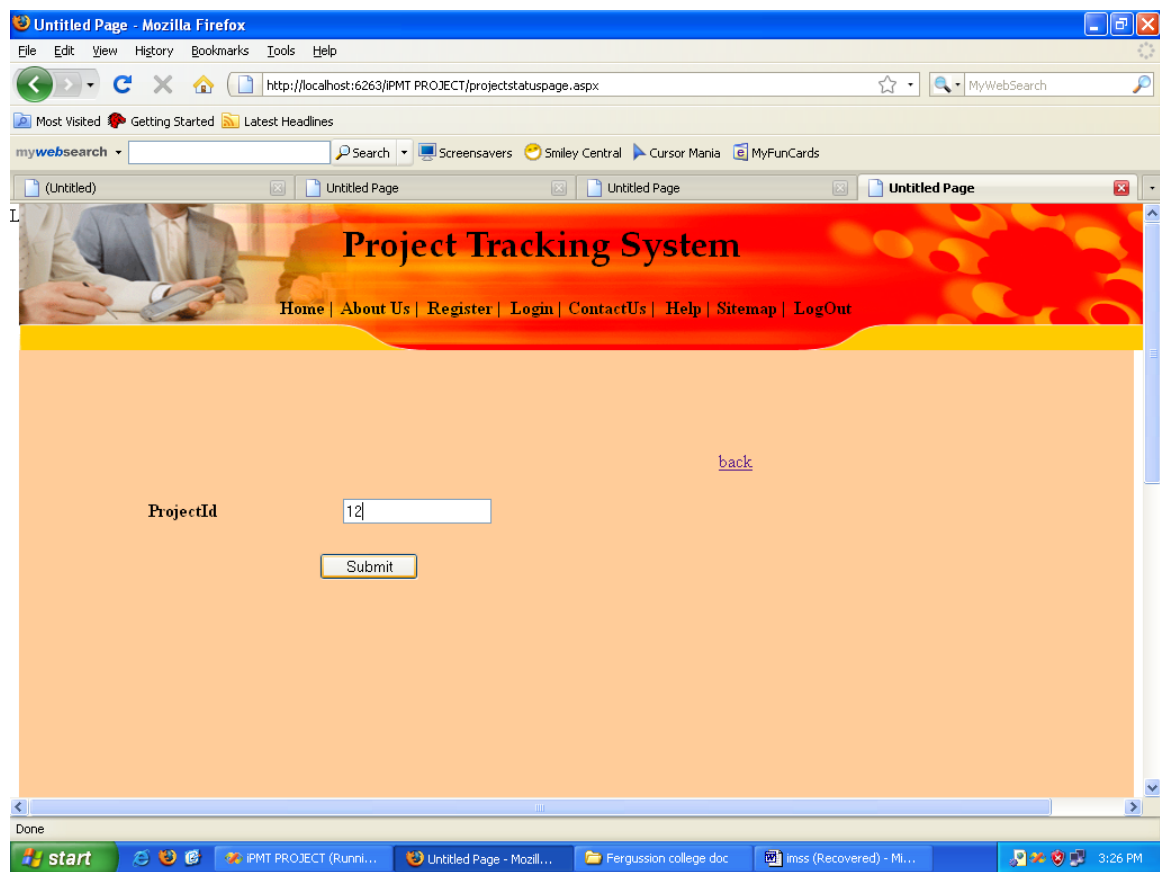
IPMT PROJECT (Runni...)

Untitled Page - Mozill...

Fergusson college doc

lms (Recovered) - M...

3:26 PM



Untitled Page - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:6263/IPMT PROJECT/viewclientdetails.aspx

MyWebSearch

Most Visited Getting Started Latest Headlines

mywebsearch Search Screensavers Smiley Central Cursor Mania MyFunCards

(Untitled) Untitled Page Untitled Page Untitled Page

Do you want Firefox to remember this password? Remember Never for This Site Not Now

Project Tracking System

[Home](#) | [About Us](#) | [Register](#) | [Login](#) | [ContactUs](#) | [Help](#) | [Sitemap](#) | [LogOut](#)

Client Id	Client Name	EmailId	Project Id	Project Desc
c001	swapna	swapna@yahoo.com	p001	cxvcvcvcvc
c002	ammu	ammu@rediff.com	p002	dsfsfsdfsdfsdfsdfsdfsdfsdfs
c003	sam	sm@hotmail.com	p003	sdfsdfsdfsdfsdfsdfsdfsdfsdfs
c004	c004	c004@gmail.com	p004	dsfdjdsdfjdfdfdfsdfsdfsdfs

1 2 3 4

[back](#)

Done

start IPMT PROJECT (Runni... Untitled Page - Mozill... Ferguson college doc mss (Recovered) - M...

3:21 PM

Untitled Page - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:6263/IPMT PROJECT/projectadddetails.aspx

Most Visited Getting Started Latest Headlines

mywebsearch Search Screensavers Smiley Central Cursor Mania MyFunCards

(Untitled) Untitled Page Untitled Page Untitled Page

HyperLink

Project Tracking System

[Home](#) | [About Us](#) | [Register](#) | [Login](#) | [ContactUs](#) | [Help](#) | [Sitemap](#) | [LogOut](#)

TeamId	123	Project Id	12
Team Size	4	Project Name	fff
Module Id	12	Client Id	123
		Project Desc	asas
Module Name	aa		
		Project Start Date	12/02/2009
Developer Name	aa	Project End Date	11/11/2009
		Project Status	aa

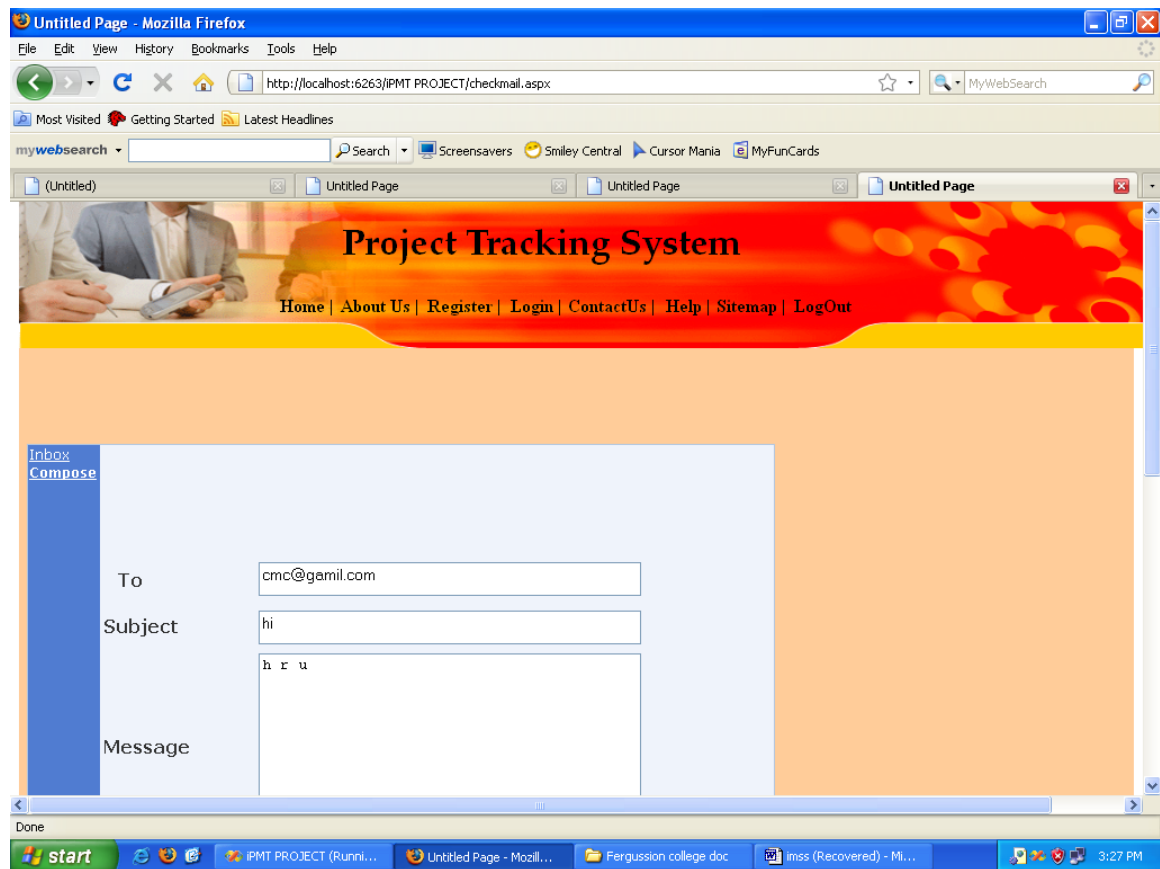
(Format: MM/DD/YYYY)

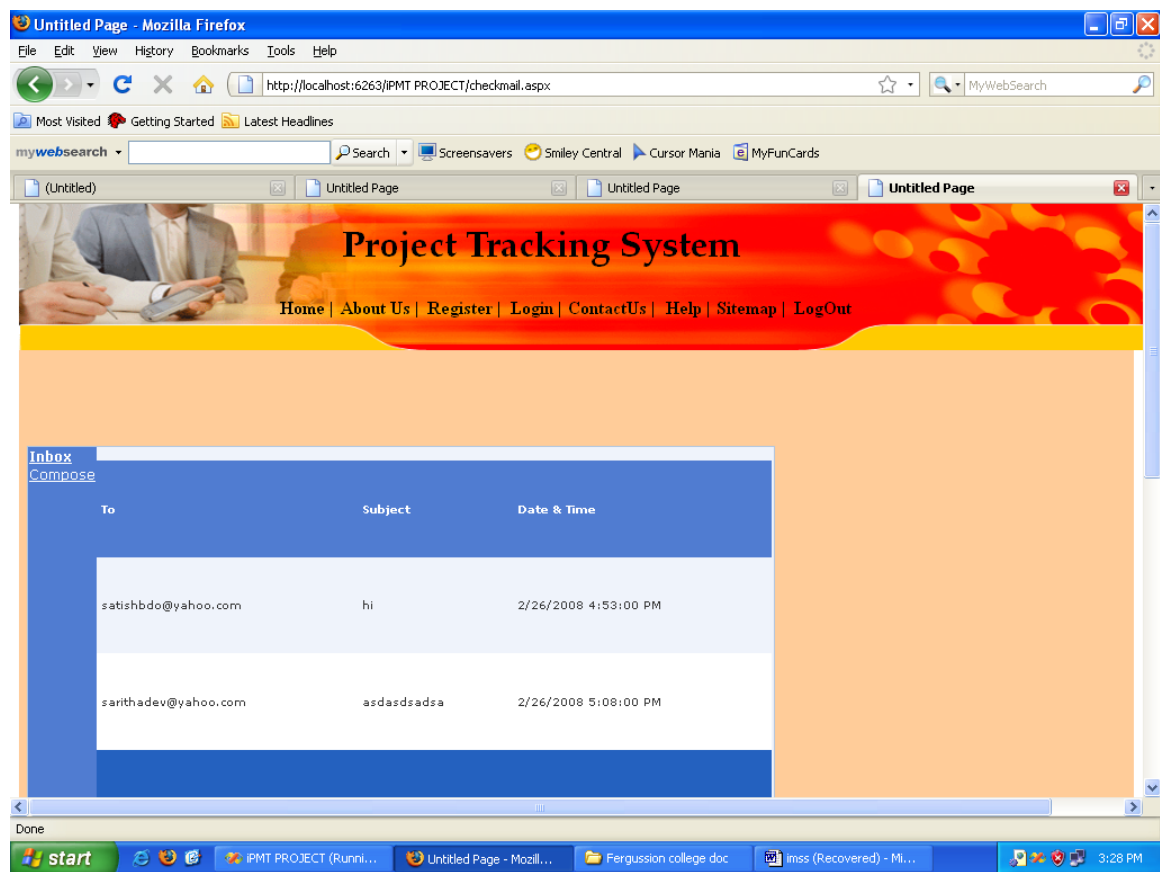
back

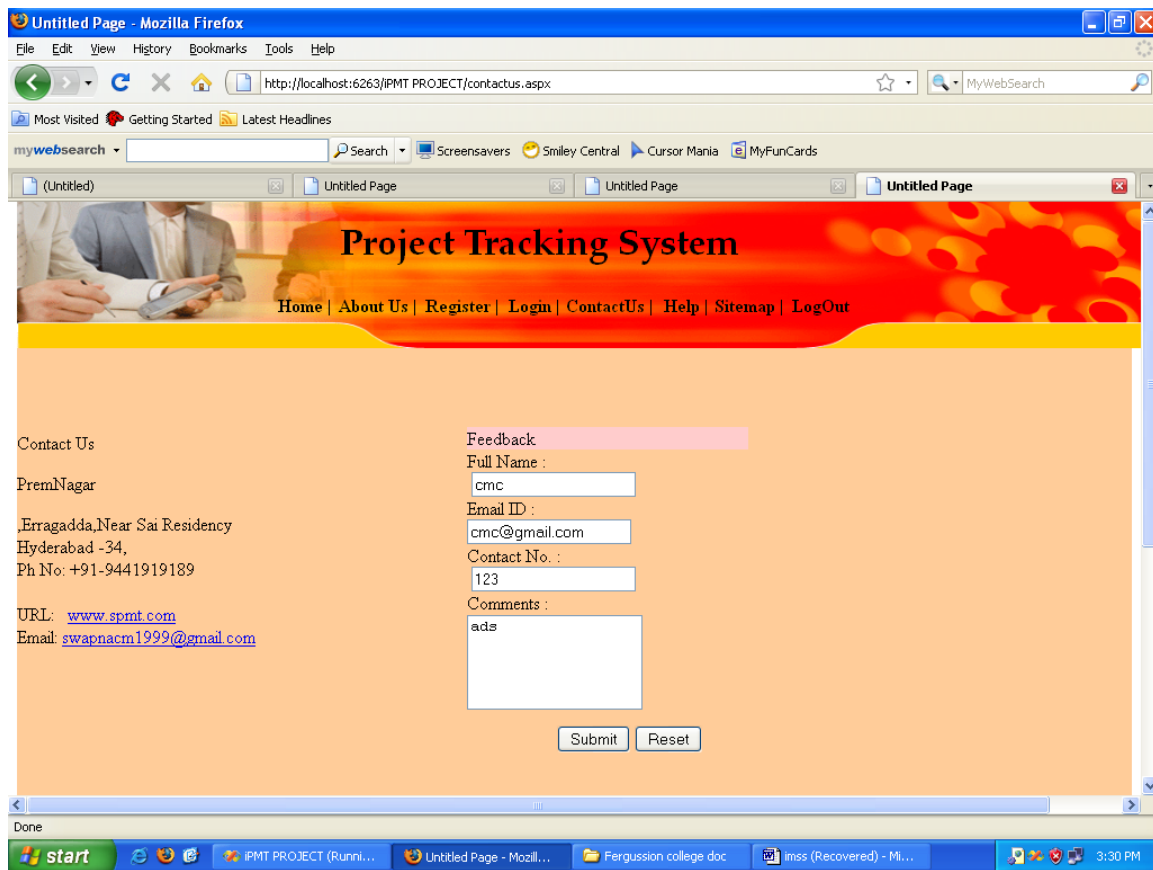
Done

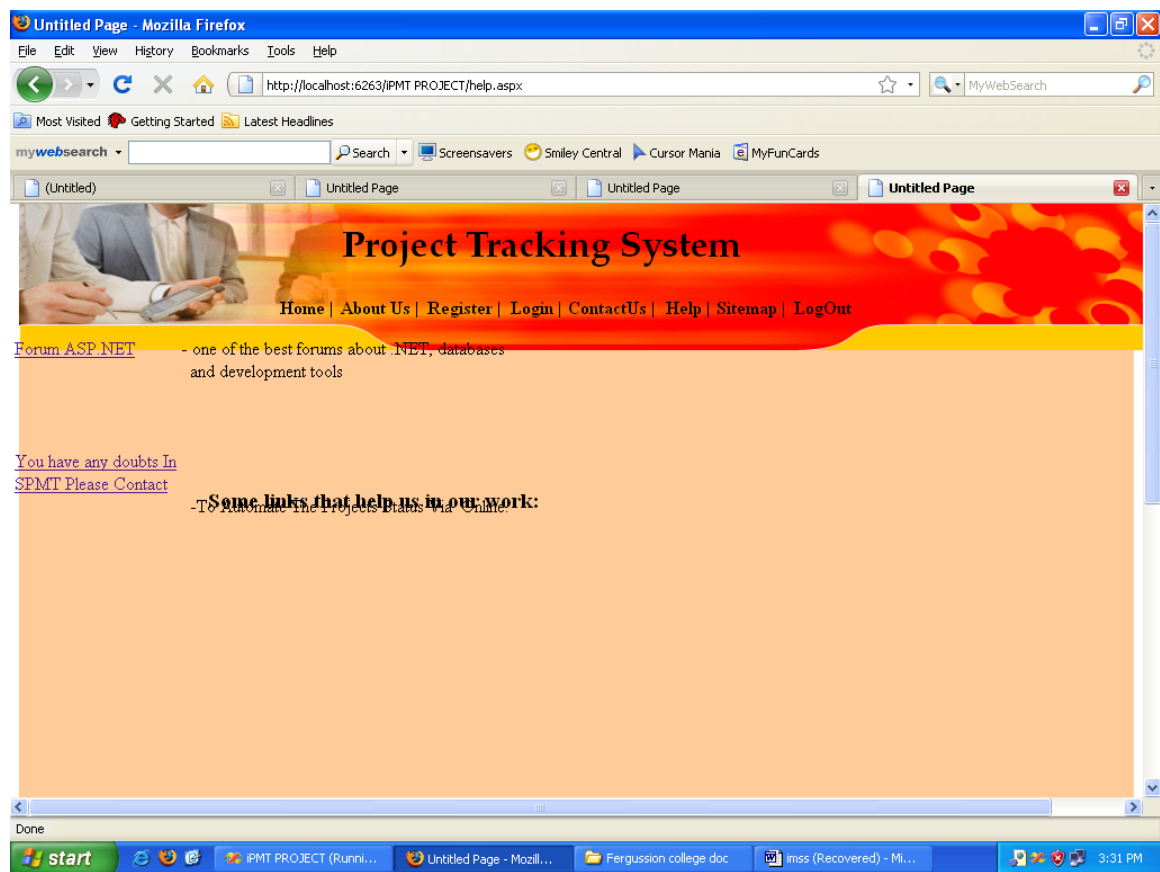
start IPMT PROJECT (Runni... Untitled Page - Mozill... Ferguson college doc mss (Recovered) - M...

3:24 PM









DESIGN CONVENTIONS

GUI STANDARDS

Labels

A label consists of read only text or graphics; it identifies components and communication the status of a process. You can use labels with a component or can use it to describe a group of components.

Text Field

A Text Field is a rectangular box that displays a single line or text. If the line is too long to fit in the text field, the text automatically Scrolls horizontally.

Password Field

A Password Field is a variation of text field. When you type in a password field, instead of characters asterisks (*) are displayed. The asterisks are referred to as a Masking agent.

List Box

A List Box is used to display a set of items. You can use a list to present user with a set of choices.

Command Buttons

The Command Button is a component with *a* rectangular box that displays a single line or text. The text typically consists of a single word that represents the action associated with that button.

SYSTEM ENVIRONMENT

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.

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 feature 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® SQL Server™ 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.

ACTIVE X DATA OBJECTS

ADO.NET Overview

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the **Connection** and **Command** objects, and also introduces new objects. Key new ADO.NET objects include the **Data Set**, **Data Reader**, and **Data Adapter**.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the **DataSet** -- that is separate and distinct from any data stores. Because of that, the **DataSet** functions as a standalone entity. You can think of the DataSet as an always disconnected record set that knows nothing about the source or destination of the data it contains. Inside a **DataSet**, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A **DataAdapter** is the object that connects to the database to fill the **DataSet**. Then, it connects back to the database to update the data there, based on operations performed while the **DataSet** held the data.

In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the **DataAdapter**, which provides a bridge to retrieve and save data between a **DataSet** and its source data store.

It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based **DataSet** object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as collections and data types. No matter what the source of the data within the **DataSet** is, it is manipulated through the same set of standard APIs exposed through the **DataSet** and its subordinate objects.

While the **DataSet** has no knowledge of the source of its data, the managed provider has detailed and specific information. The role of the managed provider is to connect, fill, and persist the **DataSet** to and from data stores. The OLE DB and SQL Server .NET Data Providers (System.Data.OleDb and System.Data.SqlClient) that are part of the .Net Framework provide four basic objects: the **Command**, **Connection**, **Data Reader** and **DataAdapter**. In the remaining sections of this document, we'll walk through each part of the **DataSet** and the OLE DB/SQL Server .NET Data Providers explaining what they are, and how to program against them.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

- **Connections**. For connection to and managing transactions against database.
- **Commands**. For issuing SQL commands against a database.
- **Data Readers**. For reading a forward-only stream of data records from a SQL Server data source.
- **Datasets**. For storing, remoting and programming against flat data, XML data and relational data.

- **Data Adapters.** For pushing data into a **DataSet**, and reconciling data against a database.

When dealing with connections to a database, there are two different options: SQL Server .NET Data Provider (System.Data.SqlClient) and OLE DB .NET Data Provider (System.Data.OleDb). In these samples we will use the SQL Server .NET Data Provider. These are written to talk directly to Microsoft SQL Server. The OLE DB .NET Data Provider is used to talk to any OLE DB provider (as it uses OLE DB underneath).

Connections

Connections are used to 'talk to' databases, and are represented by provider-specific classes such as **SqlConnection**. Commands travel over connections and result sets are returned in the form of streams which can be read by a **Data Reader** object, or pushed into a **DataSet** object.

Commands

Commands contain the information that is submitted to a database, and are represented by provider-specific classes such as **SqlCommand**. A command can be a stored procedure call, an UPDATE statement, or a statement that returns results. You can also use input and output parameters, and return values as part of your command syntax. The example below shows how to issue an INSERT statement against the **Northwind** database.

Data Readers

The **Data Reader** object is somewhat synonymous with a read-only/forward-only cursor over data. The **Data Reader** API supports flat as well as hierarchical data. A **Data Reader** object is returned after executing a command against a database.

The format of the returned **DataReader** object is different from a recordset. For example, you might use the **DataReader** to show the results of a search list in a web page.

Data Sets and Data Adapters

Data Sets

The **DataSet** object is similar to the ADO **Record set** object, but more powerful, and with one other important distinction: the **DataSet** is always disconnected. The **DataSet** object represents a cache of data, with database-like structures such as tables, columns, relationships, and constraints.

However, though a **DataSet** can and does behave much like a database, it is important to remember that **DataSet** objects do not interact directly with databases, or other source data. This allows the developer to work with a programming model that is always consistent, regardless of where the source data resides.

Data coming from a database, an XML file, from code, or user input can all be placed into **DataSet** objects. Then, as changes are made to the **DataSet** they can be tracked and verified before updating the source data. The **GetChanges** method of the **DataSet** object actually creates a second **DataSet** that contains only the changes to the data. This **DataSet** is then used by a **DataAdapter** (or other objects) to update the original data source.

The **DataSet** has many XML characteristics, including the ability to produce and consume XML data and XML schemas. XML schemas can be used to describe schemas interchanged via WebServices. In fact, a **DataSet** with a schema can actually be compiled for type safety and statement completion.

Data Adapters (OLEDB/SQL)

The **DataAdapter** object works as a bridge between the **DataSet** and the source data. Using the provider-specific **SqlDataAdapter** (along with its associated **SqlCommand** and **SqlConnection**) can increase overall performance when working with a Microsoft SQL Server databases. For other OLE DB-supported databases, you would use the **OleDbDataAdapter** object and its associated **OleDbCommand** and **OleDbConnection** objects.

The **DataAdapter** object uses commands to update the data source after changes have been made to the **DataSet**. Using the **Fill** method of the **DataAdapter** calls the SELECT command; using the **Update** method calls the INSERT, UPDATE or DELETE command for each changed row. You can explicitly set these commands in order to control the statements used at runtime to resolve changes, including the use of stored procedures. For ad-hoc scenarios, a **CommandBuilder** object can generate these at run-time based upon a select statement.

However, this run-time generation requires an extra round-trip to the server in order to gather required metadata, so explicitly providing the INSERT, UPDATE, and DELETE commands at design time will result in better run-time performance.

ADO.NET is the next evolution of ADO for the .Net Framework.

1. ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the **DataSet** and **DataAdapter**, are provided for these scenarios.
2. ADO.NET can be used to get data from a stream, or to store data in a cache for updates.

There is a lot more information about ADO.NET in the documentation.

3. Remember, you can execute a command directly against the database in order to do inserts, updates, and deletes. You don't need to first put data into a **DataSet** in order to insert, update, or delete it.
4. Also, you can use a **DataSet** to bind to the data, move through the data, and navigate data relationships

ASP.NET

ASP.NET is Microsoft's latest technology for building web-based applications and services, a successor to Active Server Pages (ASP) that draws on the power of the .NET Framework development platform and the Visual Studio .NET developer toolset. To better understand ASP.NET, it is important to understand some key concepts of the .NET development platform. It is also helpful to grasp object-oriented development (OOD), which is at the very heart of the .NET Framework that provides the foundation for ASP.NET development. It is the technology for providing dynamic and interactive web pages.

ASP.NET takes Active Server Pages (ASP) a step further to simplify the development process of web pages and enhance scalability of web applications. The conventional way of developing web applications using ASP to uncover some of the pitfalls that ASP.NET overcomes. We then discuss the benefits of ASP.NET and provide a high-level survey of the classes in the ASP.NET, such as control and page, as well as the complete syntax of ASP.NET Web Forms.

To wrap up this, we discuss the many aspects of ASP.NET development: how ASP.NET supports web services development, how to use custom server controls, and how session management has been improved to provide more scalable web solutions. This overview is designed to provide experienced developers with a solid introduction to ASP.NET and Web Forms; for additional in-depth information, see *Programming ASP.NET* by Liberty and Hurwitz (O'Reilly) and *ASP.NET in a Nutshell* by Duthie and MacDonald (O'Reilly).

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

Server-side managed code

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator.

Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service.

The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

Active Server Pages.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important **advantages** over previous Web development models:

- **Enhanced Performance.** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.

This amounts to dramatically better performance before you ever write a line of code.

- **World-Class Tool Support.** A rich toolbox and designer in the Visual Studio integrated development environment complement the ASP.NET framework. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

- **Power and Flexibility.** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers.

The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

- **Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

- **Manageability.** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

- **Scalability and Availability.** ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

- **Customizability and Extensibility.** ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

- **Security.** With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

Language Support

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

What is ASP.NET Web Forms?

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

- The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.
- The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
- The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class.

This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

An ASP.NET page can be created simply by taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required). For example, the following sample demonstrates a simple HTML page that collects a user's name and category preference and then performs a form postback to the originating page when a button is clicked.

ASP.NET provides syntax compatibility with existing ASP pages. This includes support for `<% %>` code render blocks that can be intermixed with HTML content within an .aspx file. These code blocks execute in a top-down manner at page render time.

Code-Behind Web Forms

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating .aspx file. An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

Introduction to ASP.NET Server Controls

In addition to (or instead of) using `<% %>` code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages.

Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a **runat="server"** attribute value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an **<input type="hidden">** form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>** control can be used to dynamically display rotating ads on a page.

1. ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
2. ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
3. ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.

4. ASP.NET server controls provide an easy way to encapsulate common functionality.
5. ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
6. ASP.NET server controls can automatically project both uplevel and down level HTML.
7. ASP.NET templates provide an easy way to customize the look and feel of list server controls.
8. ASP.NET validation controls provide an easy way to do declarative client or server data validation.

The .NET platform consists of four separate product groups:

Development tools and libraries

A set of languages, including C#, J#, and VB.NET; a set of development tools, including Visual Studio .NET; a comprehensive class library for building web services and web and Windows applications; as well as the Common Language Runtime (CLR). These components collectively form the largest part of the .NET Framework.

Web services

An offering of commercial web services, specifically the .NET Services initiative; for a fee, developers can use these services in building applications that require them.

Specialized servers

A set of .NET-enabled enterprise servers, including SQL Server, Exchange Server, BizTalk Server, and so on. These provide specialized functionality for relational data storage, email, and B2B commerce. Future versions of these products will increasingly support the .NET Framework.

Devices

New. NET -enabled, non-PC devices, from cell phones to game boxes.

.NET is a name for a new strategy: a blueprint for building applications for next decade. It's actually even more than that.

.NET Languages

This appendix contains two lists of languages (Microsoft-supported and third party) with compilers that generate IL to target the CLR.

A.1 Microsoft-Supported Languages for .NET

[Table A-1](#) lists commercial languages that Microsoft supports. You can find more information about each of these languages by browsing the provided URLs.

Table A-1. Microsoft-supported languages for .NET

Language	Link
Visual C# .NET	http://msdn.microsoft.com/vcsharp/
Visual J# .NET (Java)	http://msdn.microsoft.com/vjsharp/
Visual C++ .NET (Managed C++)	http://msdn.microsoft.com/visualc/
Visual Basic .NET	http://msdn.microsoft.com/vbasic/
JScript for the CLR	http://msdn.microsoft.com/workshop/languages/clinic/scripting07142000.asp

A.2 Third-Party Languages for .NET

[Table A-2](#) shows a list of third-party languages with compilers that target the CLR. Some of these are research languages, while others are commercial languages that target .NET. Browse the provided web sites to read more about the languages that interest you. As noted earlier, this list of languages could grow by the time this book hits the market, so be sure to check the following site for the most up-to-date listings: <http://www.gotdotnet.com>.

Table A-2. Third-party languages

Language	Link
APL	http://www.dyadic.com
COBOL	http://www.adtools.com/info/whitepaper/net.html
Component Pascal	http://www.citi.qut.edu.au/research/plas/projects/component_pascal.jsp
Delta Forth	http://www.dataman.ro/dforth/
Eiffel#	http://www.eiffel.com/doc/manuals/technology/dotnet/eiffelsharp/white_paper.html
Fortran	http://www.lahey.com/dotnet.htm , http://www.salfordsoftware.co.uk/compilers/ftn95/dotnet.shtml
Haskell	http://haskell.cs.yale.edu/ghc
Mercury	http://www.cs.mu.oz.au/research/mercury/dotnet.html
Mondrian	http://www.mondrian-script.org
Oberon	http://www.oberon.ethz.ch/lightning
Perl	http://www.activestate.com/ASP.NET
Python	http://www.activestate.com/ASP.NET
RPG	http://www.asna.com/pr2%5F20%5F01.asp
Scheme	http://rover.cs.nwu.edu/~scheme
Smalltalk	http://www.lessor-software.com/en/content/products/lswvst/LSWVSt.htm
S#	http://www.smallscript.net
Standard ML	http://www.research.microsoft.com/Projects/SML.NET
TMT Pascal	http://www.tmt.com/net.htm

For more information, visit the O'Reilly .NET Center at <http://dotnet.oreilly.com> and the .NET DevCenter at <http://www.oreillynet.com/dotnet>.

Choosing a Language

Choosing which language to use when developing ASP.NET applications is both easier and harder than choosing a language for classic ASP development. It is harder because it may be intimidating for some to choose between a substantially revised Visual Basic and a completely new language, C#. It is easier because the choice of language no longer requires giving up substantial amounts of functionality for your preferred language. As in many other cases, including language choice in classic ASP, a lot of the decision is determined by where you're coming from. If you're:

An experienced ASP developer who has used VBScript

You'll probably prefer Visual Basic.NET.

An experienced ASP developer who's used JScript

You'll want to look at C# or JScript.NET (keeping in mind that finding code examples in C# is easier, since the novelty of the language makes it more interesting for many).

An experienced Visual Basic developer

Visual Basic.NET is the obvious choice, but you may also find it worthwhile to check out C#, which offers a lot of the power of C++ without such a steep learning curve.

An experienced C, C++, or Java developer

You'll probably feel right at home with C#, which, as a C-derived language, shares a lot of syntax with these languages.

New to ASP.NET development, with no prior ASP experience

Visual Basic.NET will probably be easiest to learn, although C# runs a close second.

Because of the level of cross-language interoperability in .NET, your choice needn't be an either/or. You can feel free to create applications and classes in Visual

Basic.NET, C#, JScript.NET, or any .NET-enabled language, knowing that they will be able to work together smoothly and easily, thanks to the CLR.

Why and When Should We Use ASP.NET?

We should use ASP.NET for any new projects you are about to start for the following reasons:

- Reduced development time
- Increased performance
- Increased application stability
- Increased scalability
- New ASP.NET features

Some of these benefits, such as reduction in development time, assume familiarity with the .NET development platform. If you are starting your first ASP.NET development project, you should allow some time for getting up to speed on the new platform. Subsequent projects should see reduced development time over classic ASP, as developers become more familiar with the platform.

In addition to these factors, ASP.NET, like ASP, is available for free. The only costs associated with ASP.NET development are the costs of the operating system on which you wish to run your application (Windows 2000, Windows XP, or Windows Server 2003) and the cost of the development environment you choose to use. Of course, as with classic ASP, you can use free or inexpensive text editors to create your applications. Given that the .NET Framework is a free add-on to Windows (and is integrated with the Windows Server 2003 line), it is possible to create ASP.NET applications without spending a penny beyond the cost of the operating system and hardware on which it will run. Integrated development environments, such as Microsoft Visual Studio .NET 2003, are also available at an additional cost and greatly simplify ASP .NET development

Features in ASP.NET

We'll close our introductory look at the .NET platform with a list of new features that are unique to ASP.NET and the chapter in which each will be discussed.

Web Forms

A new feature that, in combination with an editor such as Visual Studio .NET provides the ASP.NET developer the same drag and drop development convenience enjoyed by Visual Basic developers for years. Web Forms improve the speed of development by encapsulating frequently used features into server controls, which are declared using a tag-based syntax similar to HTML and XML.

Web services

Web services allow developers to expose the functionality of their applications via HTTP and XML so that any client who understands these protocols can call them. Web services can make the task of application integration easier, particularly in situations in which application-to-application integration is made difficult by firewalls and/or differing platforms.

Server controls

Server controls are declared using an HTML-like syntax, making them easier to work with for page UI designers. They are executed on the server, returning HTML to the browser. Server controls may be manipulated on the server programmatically and provide power and flexibility for applications that must support a variety of browsers.

Validation

One group of server controls is designed to simplify the task of validating user input. It includes controls to validate required fields, to compare one field to another or to a specific value for validation, and to validate user input using regular expressions, which allow you to specify a format that user input must follow to be valid.

Improved security

ASP.NET offers tighter integration with Windows-based authentication, as well as two new authentication modes: forms-based authentication (which allows users to enter authentication credentials in a standard HTML form, with the credentials validated against your choice of backend credential store) and Passport authentication (which makes use of Microsoft's Passport authentication service).

Features in ASP.NET v1.1

In Version 1.1 of the .NET Framework, several features have been added that are of interest to ASP.NET developers. These include:

Request Validation

Request Validation, when enabled (the default), checks all forms of posted input (form fields, query string, etc.) and raises an exception if any HTML or script code is found. This can help prevent cross-site scripting attacks in your applications.

Side by side execution

Starting with ASP.NET 1.1, you can choose which version of the .NET Framework your application will run against. Assuming you have both Version 1.0 and Version 1.1 installed, you can configure individual applications to run against either version.

Built-in mobile control support

In Version 1.0, support for targeting mobile devices such as cell phones and PDAs was provided via a set of controls available as a separate download. In Version 1.1, these controls have been fully integrated into the .NET Framework, and a new application type has been added to Visual Studio .NET 2003 to support development of ASP.NET applications for mobile devices.

ADO.NET enhancements

In Version 1.0, developers wishing to access data from Oracle and/or ODBC data sources had to download and install a separate data provider for these data sources. In Version 1.1, the ODBC and Oracle data providers have been integrated into the .NET Framework.

ASP.NET Web Applications

The ASP.NET Web Application is the type of application most developers will work with on a regular basis. The terminology comes from the description used in the Visual Studio .NET environment to describe the project type used to create this type of application. You may also hear this type of application described as an ASP.NET Web Forms Application. For reasons we'll explore in the next chapter, we prefer the former term.

An ASP.NET Web Application, in its simplest form, consists of a directory made available via HTTP using the IIS administration tool or through the Web Sharing tab of a folder's Properties dialog (or by creating a web application project in Visual Studio .NET) and at least one ASP.NET page, designated by the .aspx file extension. This file (or files), whose structure we'll discuss in detail in the next chapter, typically contains a mix of HTML and server-side code. This HTML and server-side code combine to create the final output of the page, typically consisting of HTML markup that is sent to the client browser.

ASP.NET Mobile Web Applications

The ASP.NET Mobile Web Application is a subtype of Web Application specific to developing for mobile devices such as cell phones and PDAs. The primary thing that distinguishes a mobile web application from a standard web application in ASP.NET is the use of the ASP.NET mobile controls, which are built into the .NET Framework as of Version 1.1. These include the mobile Form control and standard controls such as labels, textboxes, and panels, as well as mobile-specific controls such as the TextView, PhoneCall, and SelectionList controls. Note that both mobile Web Forms pages (those that use the mobile controls) and standard Web Forms pages can coexist within the same application, if desired.

To simplify development of ASP.NET applications for mobile devices, Visual Studio .NET 2003 provides an ASP.NET Mobile Web Application project template. This template includes a default mobile Web Form, as well as a special

section added to the Web.config file called <deviceFilters>, which contains settings for device-specific rendering.

ASP.NET Web Services

The other type of application available to ASP.NET developers is the ASP.NET Web Service. Like ASP.NET Web Applications, there are a number of terms floating around for this type of application. (Microsoft refers to web services as "XML Web Services," perhaps in hopes of a positive association between web services and the XML standard.) A web service is an application that exposes programmatic functionality to clients over the Internet or an intranet using the underlying plumbing of a developing W3C standard called SOAP. In simple terms, it can be seen as a simple function call across the Internet.

.aspx Files

.aspx files, also known as ASP.NET pages or Web Forms, are the meat and potatoes of an ASP.NET Web Application. These files contain the HTML tags, server controls, and code that present a user interface to your users, and process their requests (or call helper functions in business-tier components to do so). Like the global.asax file, .aspx files may either contain code directly or refer to a code-behind class that contains the code for that page. Note that the code-behind used, if any, must inherit from the Page class in the System.Web.UI namespace.

.ascx Files

.ascx files are used to implement what are known as ASP.NET user controls. User controls are a technique for code reuse that lies somewhere between the function of the #Include directive in classic ASP (which you can still use in ASP.NET, if you choose) and the function of custom ASP.NET Server Controls. User controls are made up of HTML tags, server controls, and code (or any combination of the above), and can be reused through a simple tag-based syntax. They have the advantages of being simpler to develop than custom server controls, as well as offering greater functionality than includes (such as the ability to expose properties and methods).

Web Forms

Web Forms are an ASP.NET technology used to create programmable web pages. They are the primary building block of ASP.NET Web Applications. The main goal of Web Forms is to bring the same productivity to web applications that Visual Basic brought to Windows applications. Web Forms consist of the user interface

(UI) and the UI logic written on the server side. The UI and UI logic can reside either in the same file or in separate files.

Web Forms in ASP.NET offer a number of advantages over ASP and other technologies for generating web applications. ASP.NET Web Forms:

- Provide support for any HTML 3.2-compliant browser. Even ASP.NET Server Controls that provide advanced client-side functionality will gracefully degrade for browsers that do not support DHTML or script. These controls will, however, take advantage of such support in browsers such as Internet Explorer 5.0 or later.
- Are built on the Common Language Runtime and provide all the benefits of the runtime, such as managed execution, type safety, and inheritance.
- Can be built with any Common Language Runtime language, including C#, Visual Basic .NET, and JScript .NET.
- Can be created using rapid application development tools such as Visual Studio .NET. You can build a Web Forms page simply by dragging and dropping controls from the VS.NET toolbox onto the page.
- Provide a rich set of server controls that provide almost all the functionality required for a web application. ASP.NET ships with a broad array of built-in server controls.
- Offer a flexible programming model, in which code may be included in the same file as the Web Form, as in the classic ASP model or in separate module files, referred to as code-behind files. Code-behind promotes the separation of code and content, which can improve your code's readability, maintainability, and reusability.
- Preserve the state of the page and its controls between requests with the inclusion of state management features. Provide an extensible model that allows you to develop your own controls or purchase third-party controls to add functionality to your application.

Web Services

The primary purpose of ASP.NET web services is to provide access to application functionality through standard web protocols (including HTTP and XML), regardless of the application's location or the platform on which it is built. When your application exposes functionality as a web service, that functionality can be consumed by clients on any platform, presuming the clients understand XML and SOAP and can communicate via the HTTP protocol. More plainly, a web service is a function that is called over the Internet.

An ASP.NET web service can be very simple or it can provide complex functionality. It can return a variety of data types—from simple strings and integer values to complex data types such as classes and datasets.

Web services are traditionally thought of as providing only business services (e.g., you call a method, perhaps passing in some parameters, and you receive a return value), but there's no reason why you can't create a web service that returns a chunk of HTML. Doing so would allow you to provide cross-platform access to functionality similar to that provided by ASP.NET Server Controls, albeit with some performance overhead.

ASP.NET Configuration

When working with ASP.NET, you'll be called on to configure your ASP.NET applications. One major advantage that ASP.NET has over classic ASP is that most of the important configuration options for ASP.NET applications are stored in configuration files that reside in the web application directory. This makes it considerably easier to deploy an application from one server to another or to replicate an application across a web farm, since the application's configuration information will be copied along with the Web Form Pages, code-behind classes, and assemblies that make up the application.

ASP.NET Security

Security is an extremely complicated subject, and ASP.NET security is no exception. This chapter discusses the approaches you can take to secure your ASP.NET applications. Absent from the discussion are the topics of network, server, and infrastructure security. This should not be interpreted to mean these topics are unimportant. On the contrary, without properly securing any supporting servers and infrastructure, the measures you take to secure your application with the tools made available by the .NET Framework will be for naught. A discussion of these topics, however, is beyond the scope of this book. The security section of the Microsoft TechNet web site, referenced at the end of this chapter, contains a wealth of information on how to secure your servers and network properly, including tools to assist you in this important task.

The importance of securing your applications cannot be stressed enough. Failure to devote the time and resources to get security right can result in data loss, application failure or hijacking, as well as loss of revenue and/or reputation. And it's important that security be considered from the very beginning. Application security added as an afterthought is little better than no security at all.

Securing access to an application or to the resources belonging to an application involves two processes: **authentication and authorization**. This chapter explains how these processes relate to ASP.NET and how each fits into the overall scheme of allowing or preventing access to ASP.NET application resources.

The discussion focuses on the three authentication methods the ASP.NET runtime provides: Windows, Forms, and Passport. The chapter also discusses ACL-based and URL authorization, as well as strategies for obtaining secure access to data and securing web services. The discussion also touches briefly on code access security, which underlies the ASP.NET security model.

Error Handling, Debugging, and Tracing

Most code samples in this book don't include code intended to handle errors. It's not that error handling isn't important, but error handling can add complexity, and for the most part we've tried to keep the sample code as simple and clear as possible. Since you'll need to deal with errors in the real world of application programming, the first part of this chapter discusses the variety of techniques available in ASP.NET for handling errors, including custom error pages and structured exception handling—a new feature of Visual Basic .NET.

In addition to handling errors in ASP.NET applications, most developers want to figure out what's causing those errors. To that end, the latter part of this chapter discusses debugging using either the .NET Framework SDK debugger or Visual Studio .NET. The chapter also covers use of the ASP.NET trace feature to troubleshoot application problems.

Code Access Security

Code access security is a new .NET runtime feature that can dramatically reduce the likelihood of applications performing damaging actions by putting significant restrictions in place on untrusted or partially trusted code.

SQL SERVER

DATABASE

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as row or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

Relational Database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

Foreign Key

When a field in one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

Data Abstraction

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

Advantages of RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared
- Standards can be enforced
- Security restrictions ca be applied
- Integrity can be maintained

- Conflicting requirements can be balanced
- Data independence can be achieved.

Disadvantages of DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are

- The row level lock manager

Enterprise wide Data Sharing

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource.

Portability

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

Open Systems

SQL SERVER offers a leading implementation of industry –standard SQL. SQL Server's open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industries most comprehensive collection of tools, application, and third party software products SQL Server's Open architecture provides transparent access to data from other relational database and even non-relational database.

Distributed Data Sharing

SQL Server's networking and distributed database capabilities to access data stored on remote server with the same ease as if the information was stored on a single local computer. A single SQL statement can access data at multiple sites. You can store data where system requirements such as performance, security or availability dictate.

Unmatched Performance

The most advanced architecture in the industry allows the SQL SERVER DBMS to deliver unmatched performance.

Sophisticated Concurrency Control

Real World applications demand access to critical data. With most database Systems application becomes "contention bound" – which performance is limited not by the CPU power or by disk I/O, but user waiting on one another for data access. SQL Server employs full, unrestricted row-level locking and contention free queries to minimize and in many cases entirely eliminates contention wait times.

No I/O Bottlenecks

SQL Server's fast commit groups commit and deferred write technologies dramatically reduce disk I/O bottlenecks. While some database write whole data block to disk at commit time, SQL Server commits transactions with at most sequential log file on disk at commit time, On high throughput systems, one sequential writes typically group commit multiple transactions. Data read by the transaction remains as shared memory so that other transactions may access that data without reading it again from disk. Since fast commits write all data

necessary to the recovery to the log file, modified blocks are written back to the database independently of the transaction commit, when written from memory to disk.

SYSTEM TESTING

Testing is process of confirming that a program or systems does what it is proposed of. Testing is the only way to assure the quality of s/w and it is an umbrella activity rather than a separate phase. This is an activity to be performed in parallel with the s/w effort and one that consists of its own phases of analysis, design, implementation, execution and maintenance.

Testing Strategy:

Unit Testing:

This testing method considers a module as single unit and checks the unit at interfaces and communicates with other modules rather than getting into details at statement level. Here the module will be treated as BLACKBOX, which will take some inputs and generate output. Outputs for a given set of input combination are pre-calculated and are generated by the module.

Integration Testing:

Here all the pre-tested individual modules will be assembled to create a larger system and tests are carried out at system level to make sure that all modules are working in synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually and are also running cohesion with other modules. For this testing we create test-cases to check all modules once and then a generated test combination of test paths through out the system to make sure that no path is making its way into chaos.

Validation Testing:

Testing is major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests. To uncover errors that are associated with interfacing the objective is to make test modules and built a program structure that has been detected by design. In a non-incremental integration all the modules are combined in advance and the program is tested as a whole. Here error will appear in an end-less loop function. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected.

Different incremental integration strategies are

- Top-Down integration
- Bottom-Up integration
- Regression integration

Testing means quality test. Testing is a process of executing a program with the intent of finding error. A good test case is one that has a high probability of finding an as yet undiscovered error.

Objective should be to design tests that systematically uncover different classes of error and to do with a minimum amount of time and effort. Testing cannot show the absence of defects, it can only show that s/w defects are present. It is important to keep this statement in a mind as testing is being conducted.

Any engineering product can be tested in one of the two ways.

Knowing the specific function that a product has been designed to perform, test can be conducted that demonstrate each function is fully operational. This approach is called '**BLACK BOX TESTING**'

Knowing the internal working of the product, test can be conducted to ensure that "all gears mesh", that is, that internal operation of the product performs according to specification and all internal components have been adequately exercised. This approach is called "**WHITE BOX TESTING**".

These approaches provide a mechanism that can help to ensure the completeness of tests and provide the highest likelihood for uncovering errors in s/w.

The goals of verification and validation are to access and improve the quality of work products generated during development and modification of s/w. these are 2 types of verifications namely (1) life-cycle verification (2) formal verification.

Validation is the process of evaluating s/w at the end of s/w development process.

Quality assurance is a planned and systematic pattern of action necessary to provide adequate confidence that the item confirms to the technical requirement.

Walkthroughs are sessions where the material being examined is examined is presented by a review and evaluated by a team of reviewers.

Inspection involves assessing the s/w life cycle and improving the quality of work products.

Life –cycle verification is the process of determining the degree to which the work products of a given phase of the development cycle fulfill the specification established during prior phases.

Formal verification is a rigorous mathematical demonstration that source code confirms to its specification.

High quality can be achieved through testing of source code alone. Although a program should be totally free of errors, this seldom the case for large s/w products. There are 3 major categories of s/w error.

- Requirement errors
- Design errors
- Implementation errors

Quality assurance defines the objective of the project and reviews the overall activities so that the errors are corrected early in the development process.

During analysis and design, a s/w verification plan and acceptance test plan is prepared. The verification plan describes the methods to be used in verifying that the requirements are satisfied by the design documents and that the source is consistent with the requirements specification and design documents. The acceptance test plan includes test cases, outcomes and capabilities demonstrated by each test case. Following completion of the verification plan and acceptance plan, an s/w verification review is held to evaluate the adequacy of the plans.

During product evolution, in-process audits are conducted to verify consistency and completeness of the work products. Items to be audited for consistency include interface specification for h/w, s/w and people; internal design verses functional requirements verses test descriptions.

Prior to product delivery, a functional audit and a physical audit are performed. The functional audit reconfirms that all the requirements have been met. The physical audit verifies that the source code and all associated documents are complete, consistent with one another and ready to deliver. An s/w verification summary is prepared to describe the results of all reviews.

SYSTEM TESTING:

A system is tested for online responses, volume of transactions, stress, recovery from failure and usability. System testing involves two kinds of activities –integration testing and acceptance testing.

ACCEPTANCE TESTING:

It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

1. Testing coverage analyzer-records the control paths followed for each test case.
2. Timing analyzer-also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.
3. Coding standards-static analyzers and standard checkers are used to inspect code for deviations from standards and guidelines.

ALPHA and BETA TESTING:

If s/w is developed as product to be used by many customers, it is impractical to perform formal acceptance test with each one. So, most developers use Alpha and Beta testing to uncover errors that only the end user seems able to find.

Alpha testing is conducted by the customer in the presence of my project leader and recorded the errors and usage problems what they faced.

Beta testing is conducted at customer site by the end users of the s/w. the customer recorded all problems that are encountered during beta testing and sent those problems to us regular intervals. Then we made the modification and released to the entire customer base.

CONCLUSION

All the objectives that had been charted out in the initial phases were achieved successfully.

System Features:

System satisfies all the requirements for which the company developed the system. System has strong security. System is fully GUI based. It is easy to operate and user-friendly. Platform includes the in built back-up and recovery facility.

Working on the project was a good experience. Working together in teams helped us to communicate better. We understand the importance of planning and designing as a part of software development.

The concept pf peer-reviews helped to rectify the problems as and when they occurred and also helped us to get some valuable suggestions that were incorporated by us. Developing the project has helped us to gain some experience on real time development procedures.

Suggestion for further development:

System required sufficient RAM and faster server to run the application properly.

BIBLIOGRAPHY

1. ASP.Net : ASP.Net Unleashed
2. SQL Server 2000 : SQL Unleashed
3. <http://www.msdn.microsoft.com/webservices>

