****

INTRODUCTION

**PROJECT DETAIL**

**INTRODUCTION**

The emart is the part of the sample application that provides customers with online shopping. Through a Web browser, a customer can browse the catalog, place items to purchase into a virtual shopping cart, create and sign in to a user account, and purchase the shopping cart contents by placing an order with a credit card.

By integrating information and improving processes, emart will help improve enterprise-wide decision support and operational efficiency. Improved efficiency translates into:

* More resources to support the enterprise .
* Improved customer service
* Web-based and more user friendly interfaces to buyers .

An extremely powerful marketing tool, emart’s ability to suggestively sell alternate and complementary products; e.g. if a customer selects the "standard" version of a product, you can recommend that they also consider the "pro" version or if your customer chooses a pair of pants, you can suggest that they also consider a belt and socks. Simply specify related product skus and optional text in a product’s definition in your Merchant Administrator, and PDG Shopping Cart will up-sell or recommend related products to your customers automatically

PROJECT DESCRIPTION

DESCRIPTION OF PROJECT

The category screen shows all of the products available for a particular category.

The product screen shows all of the items in a particular product. Product screen’s right column of the list shows the price of the item, and includes a link labelled Add to Cart. This link, which also appears on screen for the corresponding item, allows the customer to add the item to the cart without looking at the item details

The item screen shows detailed information about an individual item for sale .

The Add to Cart link, when clicked, adds an order for the item to the shopping cart, and then shows the shopping cart contents

The cart screen lists the items currently in the cart, allows the customer to change the quantity of each item ordered, and shows a title. It also includes a link to remove the item from the cart, and a link Proceed to Checkout which, when clicked, shows the order information screen if the user is signed on. If the user is not signed on, the signon screen is shown instead.

The signon screen allows an existing customer to sign in as an existing user, and a new customer to create an account. An existing customer enters a user name and password, and the application displays the Order Information screen shown in . A new customer enters a user name and password and clicks the button Create New Account. The application creates a user with the requested password. If user creation succeeds, the application displays the Account Information screen.

The account information screen, shown in collects information about a new customer, including contact information, a credit card, and personal preferences. This is also the screen displayed whenever the customer clicks the Account link at the top right corner of the screen (beneath the Search box). Clicking the Update button directs the browser to a page that summarizes the information entered.

The Order Information screen allows the user to enter billing and shipping address. Default values for the addresses come from the contact information for the currently signed-in customer. This information is transmitted to the application when the user clicks Submit. The application creates a new order, sends it to the order processing center, and displays the Order Complete screen.

The Order Complete screen verifies to the user that the order has been placed. The screen includes the order number

**Key Strengths**

A steady stream of innovative features born of a

deep understanding of online shopping as a social

experience. Recent enhancements and additions

include:

g The beginnings of social networking

capabilities, with customers able to connect

with and keep tabs on a network of friends

g The ability for users to “tag” products by

assigning them keywords

g Personal blogs for customers

g Product discussion boards

g The ability to upload images as visual reviews

and product commentary

OBJECTIVE

**OBJECTIVES**

**Our Objective**

Our objective was to look at online retail from a customer’s point-of-view and determine which

of the biggest US-based ecommerce sites was providing the best customer experience.

A secondary aim was to zero in on key trends and identify opportunities for high-impact

customer experience improvements.

**Our Approach**

**We examined the entire experience.** Unlike some studies that focus only on browsing,

we looked at the entire experience from the first moment on the site through selection,

purchase, shipping and returns.

**We combined objective measurement with expert subjective analysis.** Metrics

have a strong appeal, but in something that involves as many intangibles as the customer

experience measurement can only take you so far. With this in mind, we split our analysis

between straightforward objective grading and the expert opinions of our panelists. (In

poker terms, think of a good objective score as a table stakes and a good subjective

score as a hand that could win you the game.)

**We analyzed a large pool of sites defined by a third party.**

**Objective evaluation criterion**

the interface must use language that is easily understood by target users .

the interface must use graphics or icons that are easily understood by target

users .

the text must be easy to read .

navigation elements must be easily recognized and understood .

product categories must be well thought out, and intuitive .

there must be effective decision support tools to guide you through the exploration

process .

the product must be clearly displayed with appropriate visualization tools .

If customer I have been to the site before, it must remember me and personalize customer’s

subsequent experience?

it is easy to keep track of things customer want to buy .

customer need to register in order to make a purchase or access certain

content .

it does have a history of customer’s transactions that he can easily access .

****

SYSTEM DEVELOPMENT

LIFE CYCLE

#### System Development Methodology

System Development Life Cycle

#### PHASES:

#### 1.2.1      Initiation Phase

The initiation of a system (or project) begins when a business need or opportunity is identified. A Project Manager should be appointed to manage the project. This business need is documented in a Concept Proposal. After the Concept Proposal is approved, the System Concept Development Phase begins.

#### 1.2.2      System Concept Development Phase

Once a business need is approved, the approaches for accomplishing the concept are reviewed for feasibility and appropriateness. The Systems Boundary Document identifies the scope of the system and requires Senior Official approval and funding before beginning the Planning Phase.

#### 1.2.3      Planning Phase

The concept is further developed to describe how the business will operate once the approved system is implemented, and to assess how the system will impact employee and customer privacy. To ensure the products and /or services provide the required capability on-time and within budget, project resources, activities, schedules, tools, and reviews are defined. Additionally, security certification and accreditation activities begin with the identification of system security requirements and the completion of a high level vulnerability assessment.

#### 1.2.4      Requirements Analysis Phase

Functional user requirements are formally defined and delineate the requirements in terms of data, system performance, security, and maintainability requirements for the system. All requirements are defined to a level of detail sufficient for systems design to proceed. All requirements need to be measurable and testable and relate to the business need or opportunity identified in the Initiation Phase.

#### 1.2.5      Design Phase

The physical characteristics of the system are designed during this phase. The operating environment is established, major subsystems and their inputs and outputs are defined, and processes are allocated to resources. Everything requiring user input or approval must be documented and reviewed by the user. The physical characteristics of the system are specified and a detailed design is prepared. Subsystems identified during design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each software module.

#### 1.2.6      Development Phase

The detailed specifications produced during the design phase are translated into hardware, communications, and executable software. Software shall be unit tested, integrated, and retested in a systematic manner. Hardware is assembled and tested.

#### 1.2.7      Integration and Test Phase

The various components of the system are integrated and systematically tested. The user tests the system to ensure that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. Prior to installing and operating the system in a production environment, the system must undergo certification and accreditation activities.

#### 1.2.8      Implementation Phase

The system or system modifications are installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

#### 1.2.9      Operations and Maintenance Phase

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements, and needed system modifications are incorporated. The operational system is periodically assessed through In-Process Reviews to determine how the system can be made more efficient and effective. Operations continue as long as the system can be effectively adapted to respond to an organization’s needs. When modifications or changes are identified as necessary, the system may reenter the planning phase.

#### 1.2.10      Disposition Phase

The disposition activities ensure the orderly termination of the system and preserve the vital information about the system so that some or all of the information may be reactivated in the future if necessary. Particular emphasis is given to proper preservation of the data processed by the system, so that the data is effectively migrated to another system or archived in accordance with applicable records management regulations and policies, for potential future access.

**SDLC Objectives**

This guide was developed to disseminate proven practices to system developers, project managers, program/account analysts and system owners/users throughout the DOJ. The specific objectives expected include the following:

* To reduce the risk of project failure
* To consider system and data requirements throughout the entire life of the system
* To identify technical and management issues early
* To disclose all life cycle costs to guide business decisions
* To foster realistic expectations of what the systems will and will not provide
* To provide information to better balance programmatic, technical, management, and cost aspects of proposed system development or modification
* To encourage periodic evaluations to identify systems that are no longer effective
* To measure progress and status for effective corrective action
* To support effective resource management and budget planning
* To consider meeting current and future business requirements

**Key Principles**

This guidance document refines traditional information system life cycle management approaches to reflect the principles outlined in the following subsections. These are the foundations for life cycle management.

**Life Cycle Management Should be used to Ensure a Structured Approach to Information Systems Development, Maintenance, and Operation**

This SDLC describes an overall structured approach to information management. Primary emphasis is placed on the information and systems decisions to be made and the proper timing of decisions. The manual provides a flexible framework for approaching a variety of systems projects. The framework enables system developers, project managers, program/account analysts, and system owners/users to combine activities, processes, and products, as appropriate, and to select the tools and methodologies best suited to the unique needs of each project.

**Support the use of an Integrated Product Team**

The establishment of an Integrated Product Team (IPT) can aid in the success of a project. An IPT is a multidisciplinary group of people who support the Project Manager in the planning, execution, delivery and implementation of life cycle decisions for the project. The IPT is composed of qualified empowered individuals from all appropriate functional disciplines that have a stake in the success of the project. Working together in a proactive, open communication, team oriented environment can aid in building a successful project and providing decision makers with the necessary information to make the right decisions at the right time.

**Each System Project must have a Program Sponsor**

To help ensure effective planning, management, and commitment to information systems, each project must have a clearly identified program sponsor. The program sponsor serves in a leadership role, providing guidance to the project team and securing, from senior management, the required reviews and approvals at specific points in the life cycle. An approval from senior management is required after the completion of the first seven of the SDLC phases, annually during Operations and Maintenance Phase and six-months after the Disposition Phase. Senior management approval authority may be varied based on dollar value, visibility level, congressional interests or a combination of these.

The program sponsor is responsible for identifying who will be responsible for formally accepting the delivered system at the end of the Implementation Phase.

**A Single Project Manager must be Selected for Each System Project**

The Project Manager has responsibility for the success of the project and works through a project team and other supporting organization structures, such as working groups or user groups, to accomplish the objectives of the project. Regardless of organizational affiliation, the Project Manager is accountable and responsible for ensuring that project activities and decisions consider the needs of all organizations that will be affected by the system. The Project Manager develops a project charter to define and clearly identify the lines of authority between and within the agency’s executive management, program sponsor, (user/customer), and developer for purposes of management and oversight.

**A Comprehensive Project Management Plan is Required for Each System Project**

The project management plan is a pivotal element in the successful solution of an information management requirement. The project management plan must describe how each life cycle phase will be accomplished to suit the specific characteristics of the project. The project management plan is a vehicle for documenting the project scope, tasks, schedule, allocated resources, and interrelationships with other projects. The plan is used to provide direction to the many activities of the life cycle and must be refined and expanded throughout the life cycle.

**Specific Individuals Must be Assigned to Perform Key Roles Throughout the Life Cycle**

Certain roles are considered vital to a successful system project and at least one individual must be designated as responsible for each key role. Assignments may be made on a full- or part-time basis as appropriate. Key roles include program/functional management, quality assurance, security, telecommunications management, data administration, database administration, logistics, financial, systems engineering, test and evaluation, contracts management, and configuration management. For most projects, more than one individual should represent the actual or potential users of the system (that is, program staff) and should be designated by the Program Manager of the program and organization

****

SYSTEM STUDY AND PROBLEM FORMULATION

**REQUIRMENTS ANALYSYS**

**SYSTEM STUDY AND PROBLEM FORMULATION**

**THE EXISTING SYSTEM**

The existing system work manually. The existing system has got lot of intricacies within itself and need lot of human effort and paper works. All above the data need to be maintained on ledgers and maintaining this is a tedious and risky process. As the transactions increases, so the data too. So the task of maintaining them increases exponentially. To view a data may need lot of paper to be searched.

Some of the negative aspects of the existing system are as follows:

1. Course of action is time consuming. Wherever a need for search arises, the process evolves search through paper records.
2. Readability of records is constrained. All the records may not be handled or written by the same person. So the format and style of records differ and hence it is difficult to understand.
3. Paper records are easily damaged in course of time. The life time of paper record is unreliable less that it easily gets spoiled.
4. Expenditure is high. Manual system needs added man power.
5. Prone to corruption by unauthorized users. Securing of manual system is not fully guaranteed. Inaccuracy can be expected. Data can get easily scratched.
6. Techniques used are more complicated. Proper techniques are not exposed, so the functioning is intricate.

To overcome these, the proposed system has been suggested.

**THE PROPOSED SYSTEM**

The proposed system is a computerized one. This has greater accuracy and efficiency. This takes only limited time for calculation.

The proposed system can be used to maintain efficiently the HR Department schedule of any type of company. In larger organizations employees are large. At that time also the proposed system is useful and helpful. HR Management System is not only become a desire of the company but it become the need of the company.

The system includes two users.

1. Administrator
2. General User

The user gets into the system using user name and a unique password. Each user has his own accessibility permission to accomplish his task flawlessly.

Advantages of the proposed system are

1. Easy access to the data
2. The new system is more user friendly, reliable and flexible.
3. Pop-up menus to carry out transactions for a new user, and for other alert messages.
4. Timely Report generation.

## Requirement Analysis

At the heart of system analysis is a detailed understanding of all important facets of business area under investigation. (For this reason, the process of acquiring this is often termed the detailed investigation) Analyst, wirking closely with the employees and managers, must study the business process to answer

these key questions:

* What is being done?
* How is it being done?
* How frequent does it occur?
* How great is the volume of transaction or decisions?
* How well is the task being performed?
* Does a problem exist?
* If a problem exist, how serious is it?
* If a problem exists, what is the underlying cause?

Requirement analysis relies on fact-finding techniques. These include:

* Interview
* Questionnaires
* Record inspection
* On-site observation

**IDENTIFICATION OF NEED**

In the world we are growing for globalization day by day with the development in IT resources and advancement, by using latest technologies every organization wants to beat its competitors and want to grow. Enterprise Resourceful Planning is the need of today’s organization. Survival on manual system is difficult so, that’s why organization of the corporate world wants to computerize their departments. The modules should be complete database driven and interactive that should provide the proper information about the Placement and Training Organization.

Success of any system depends up to a large extent on how accurately a problem is defined, thoroughly investigated and properly carried out to the choice of solution. Analysis is the only phase in which the requirements for the new system are identified. System analysis is a detailed study of the various operations performed by a system and their relationship within and outside of the system. The question is: what must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related system. During analysis data are collected on the available files, decision points and transactions handled by the parent system. Data flow diagram, interviews, onsite observations, questionnaires are used as a logical system model and tools to perform the analysis.

An analyst is responsible for performing following tasks:

* Gathered all facts about the present system from the employees.
* Studied strength and weakness of the current system.
* Determined “what” must be done to solve the problem.
* Prepared a functional specifications document.

In order to reduce the time, there is a need for computerized system that cans retrieve data, insert data, update existing data or delete existing data. These modules are developed wit the aim of reducing time, reducing manpower, reducing cost so that the records can be easily maintained. The volume of work and complexity are increasing year by year. This system reduces complexity and workload.

**PRELIMINARY INVESTIGATION**

A request to take assistance from information system can be made for many reasons, but in each case some one in the organization initiate the request. When the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

* Request clarification
* Feasible Study
* Request approval

Many requests from employees and users in the organization are not clearly defined. Therefore, it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

The feasibility study is carried out by a small group of people who are familiar with information system techniques, understand the parts of the business or organization that will be involved or affected by the project, and are skilled in the system analysis and design process.

* **Request Approval:**

It is not necessary that all request projects are desirable or feasible. Some organizations receive so many projects request from employees that only a few of them can be purchased. However, those projects that are feasible and desirable should be put into a schedule.

In some cases, development can start immediately, although usually system staff members are busy on other ongoing projects. When such situation arises, management decides which projects are more urgent and schedule them accordingly. After a project request is approved, its cost, priority, completion time and personal requirements are estimated and used to determine where to add it to any existing project list. Later on, when the other projects have been completed, the proposed application development can be initiated.

Analysis is a process of studying a problem and to find the best solution to that problem. System analysis gives us the target for the design and the implementation. Analysis is one phase, which is important phase for system development lie cycle. System development is a problem solving techniques. Analysis involves interviewing the client and the user. Three people and the existing documents about the current mode of operation are the basic source of information for the analyst.

Analysis is the process of studying a problem to find the best solution to that problem. System analysis gives us the target for the design and the implementation. Analysis is one phase of the very important phase of the system development life cycle. System development is a problem solving techniques. Analyses involve interviewing the client and the user. These people and the existing document about the current mode of operation are the basic source of information for the analyst.

Human Resource Department of the organization controls the manpower planning by recording manpower requirements, grade, job specifications and presently vacancies and strength. It maintains the candidate’s databank for the respective post.

**SOFTWARE ENGINEERING PARADIGM**

Computer Aided Software Engineering can be as simple as a single tool that support a specific software engineering activity or as complex as a complete “environment “ that encompasses tools, a database, people, hardware, a network, operating system, standards, and myriad other components. Each building block forms a foundation for the next, with tools sitting as the top of the heap. It is interesting to note that the foundation for effective CASE environment has relatively little to do for software engineering tools themselves. Rather, successful environments appropriate hardware and systems software. In addition, the environment architecture must consider the human work patterns that are applied during the software engineering process.

The environment composed of the hardware platform and system support (including networking software, software management, and object management services), the groundwork for CASE. But the CASE environment itself demands other building blocks. A set of portability services provides a bridge between CASE tools and their integration framework and the environment architecture. The integration framework is a collection of specialized programs that enables individual’s CASE tools to communicate one another, to create a project database, and to exhibit the same look and feel to the end user (the software engineer). Portability services allow CASE tools and their integration framework to migrate across different hardware platforms and operating system with out significant adaptive maintenance.

The building blocks represent a comprehensive foundation for the integration of CASE tools. However, most CASE tools in use today have been: constructed using all these building blocks.

In fact some CASE tools remain “point solution:” That is, a tool is used to assist in a particular software engineering activity (e.g. analysis modeling) but does not directly communicate with other tools, is not tied into a project database, is not part of an integrated CASE environment (l-CASE). Although this situation is not ideal, a CASE tool can be used quite effectively, even if it is a point solution.

At the low end of the integration spectrum is the individual (point solution) tool. When individual tools can provide tools provide facilities for data exchange, the integration level is improved slightly. Such tools produce output in a standard format that should be compatible with other tools that can read the format. In some cases, the builder of complementary CASE tools work together to form a bridge between the tools (e.g. an analysis and design tool that is coupled with a code generator). Using this approach, the team developed, synergy between the tools separately. Single source integration occurs when a single CASE tools vendor integrates a number of different tools and sells them as a package.

Although this approach is quite effective, the closed architecture of most single source environments precludes easy addition from other vendors.

PROJECT CATEGORY

**PROJECT CATEGORY**

Category of this project is RDBMS based, n-tier architecture, Distributed environment project with server-side components.

Project can be categorized in two ways:-

* **Local Area Network projects.**
* **Distributed Projects**.

Local Area Network projects are those projects where application has to be incorporated in the Local area network of the client i.e within its premises only. In case of LAN, server is not remotely located and client accesses the application through the network. Here the question of platform independence does not arise and we can use the technologies like: Visual Basic, Fox pro, D2k or C, C++, etc.

Distributed projects are those projects where application is remotely situated. In these kinds of projects, application is remotely situated on to the remote server from where client machine accesses the application. WAN and Internet is a kind of distributed application where client machine connects to the remote server and application is downloaded on to the client machine. Here the question of platform independence arises and we use technologies like Java Servlet, Java Server Pages, Java Beans, RMI, etc.

SOFTWARE REQUIRMENT SPECIFICATION

**SOFTWARE REQUIREMENT SPECIFICATION**

Software requirement specification (SRS) is the starting point of the software development activity. Little importance was given to this phase in the early days of software development. The emphasis was first on coding and then shifted to design.

As systems grew more complex, it became evident that the goals of the entire system cannot be easily comprehended. Hence the need for the requirement analysis phase arose. Now, for large software systems, requirements analysis is perhaps the most difficult activity and also the most error prone.

Some of the difficulty is due to the scope of this phase. The software project is initiated by the client’s needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify the requirements by talking to these people and understanding their needs. In situations where the software is to automate a currently manual process, most of the needs can be understood by observing the current practice.

The SRS is a means of translating the ideas in the minds of the clients (the input), into formal document (the output of the requirements phase). Thus, the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input has none of these properties.

PLATEFORM (Technologies/Tools)

**PLATEFORM (Technology/Tool Selection)**

**Introduction to Java/.net**

Java is a high-level, third-generation programming language like C, FORTRAN, Perl and many others. It is a platform for distributed computing – a development and run-time environment that contains built-in support for the World Wide Web.

**History of Java**

Java development began at Sun Microsystems in 1991, the same year the World Wide Web was conceived. Java’s creator, James Gosling did not design java for the Internet. His Objective was to create a common development environment for consumer electronic devices which was easily portable from one device to another. This effort evolved into a language, code named Oak and later renamed Java that retains much of the syntax and power of c++, but is simpler and more platform-independent.

**Java Features**

Some of the important features of Java are as follows:

* Simplicity
* Orientation
* Platform Independence
* Security
* High Performance
* Multi Threading
* Dynamic linking.
* Garbage Collection.

One of the most important features of Java is platform independence, which makes it famous and suitable language for World Wide Web.

**Why Java is Platform Independent?**

Java is Platform Independent because of Java Virtual Machine (JVM).

**Java Virtual Machine (JVM)**

The client application or operating system must have a java byte-code interpreter to execute byte-code instructions. The interpreter is a part of a larger program called the JVM. The JVM interprets the byte code into native code and is available on platforms that support Java.

When the user runs a Java program, it is up to the JVM to load, possibly verify, and then execute it. The JVM can perform this function from within a browser or any other container program or directly on top of the operating system.

When a browser invokes the JVM to run a Java program, the JVM does a number of things:

* It validates the requested byte-code, verifying that they pass various formatting and security checks.
* It allocates memory for the incoming java class files and guarantees that the security of JVM is not violated. This is known as the class loader module.
* It interprets the byte code instructions found in the class files to execute the program.

**Connectivity using JDBC**

There are four kinds of drivers available in Jdbc: -

* **Jdbc-Odbc Bridge Driver**
* **Partly Java Driver**
* **Native Driver**
* **Pure Java Driver**

**Jdbc-Odbc Driver:**

This provides a bridge between the Jdbc APIs and the Odbc APIs. The bridge translates the standard JDBC calls to corresponding ODBC calls, and sends them to the ODBC data source via ODBC libraries. This configuration requires every client that will run the application to have the JDBC-ODBC bridge API, the ODBC driver and the native language-level APIs. As a result this kind of driver is most appropriate on a corporate network where client installations are not a major problem.

**Partly Java Driver:**

Jdbc database calls are translated into vendor-specific API calls. The database will process the request & send the result back through the API, which in turn forwards them back to the Jdbc drivers. The Jdbc driver translates the result to the Jdbc standard & returns them to the Java application, hence this kind of driver has same problem that was with Jdbc-Odbc driver and is mostly used in Intranet.

These are true 100% pure java real JDBC drivers. All the mechanism of the client access is coded completely in java. There are no calls out off or into the virtual machine and native code and there is no need for some costly server in the middle. Type 4 drivers are different for different RDBMS and are available for almost all major RDBMS vendors.

**Pure Java Driver:**

This is a platform independent driver as this kind of driver remains on server. This kind of driver is provided by third party vendor. This middle-ware server is able to connect its java clients to many different databases. These are really non-drivers. They are front end for database access servers and connectors. For ex: The proxy driver talks to the middle tier concentrator or access server. The concentrator or access server in turn uses ODBC (or) vendor specific protocol to talk to the actual database. The requirement for collaborating middle tier server is often cumbersome and very expensive too.

**Native Driver:**

This kind of driver converts JDBC calls into the network protocol used by Database directly. This allows a direct call from client machine to the Database server.

These are true 100% pure java real JDBC drivers. All the mechanism of the client access is coded completely in java. There are no calls out off or into the virtual machine and native code and there is no need for some costly server in the middle. Type 4 drivers are different for different RDBMS and are available for almost all major RDBMS vendors.

**Client Side Interface:**

In client side interface we are using:-

* Servlet / JSP – for Internet Based Application.

Servlet / JSP are middle-ware technologies which are used in web based projects because they use:-

* HTTP Protocol to handle Request and Response.
* They are invoked through Browser.
* They give output in HTML format.
* They need Browser Support.

**ABOUT J2EE**

# Introduction to J2EE:

The multi-tier architecture such as COBRA has got its own advantages in terms of scalability, performance and reliability.

In a multi-tier architecture, a client does not interact directly with the server. Instead, it first contacts another layer called Middleware. The middleware instantiates the server applications and messages the server object. It returns results to the clients. The presence of a middleware layer allows programmers to concentrate on business logic of application. The middleware handles low-lever services, such as thread handling, security, and transactions management.

Sun Microsystems introduced the J2EE application server and the enterprise Java Bean (EJB) specifications as a venture into the multi-tier component architecture. J2EE functions as a middle tier server in three tier architectures.

It provides certain specifications that can be used to implement enterprise solutions for certain all types of business requirements. J2EE also offers cost effective solution for business solution.

J2EE is used for developing, deploying and executing applications in a distributed environment. The J2EE applications server acts as a platform for implementing various server side technologies Servlets, Java Server Pages (JSP) and Enterprise Java Bean (EJB). J2EE allows you to focus on your business logic program. The business logic is coded in java program, which are reusable component that can be accessed client program EJB runs on J2EE server.

In J2EE security is handled almost entirely by platform and its admin. The developer does not have to worry about writing the security logic.

* **J2EE Architecture:**

The J2EE SDK architecture consists of the following components:

* **The J2EE server**
* **The EJB Container**
* **The Web Container**

The J2EE server provides the EJB and web containers. The J2EE server enforces authenticating users. The either service provided by the J2EE server are listed here below.

* It allows client to interact with Enterprise Bean.
* It enables a web browser to access servlets and JSP files
* It provides naming and directory services to enable users and various services to locate and search for services and components.

The EJB container manages the execution of Enterprise Bean for J2EE server. EJB is a specification for making server side component that enable and simplifies the task of creating distributed objects. EJB component provide services such as transaction and security management and can be customized during deployment.

The web container manages the executing of JSP and servlets for J2EE applications web components and their container run on the J2EE server. Servlets of the java program that can be deployed on a java enable web server to enhances and extend the functionality of the web server for example you can write a servlets to add a manager service to a website.

Servlet can also be used to add dynamic content to web pages. Java Server Page (JSP) adds server side programming functionality to java. JSP consists of regular Html tags representing the static content and code enclosed within special tags representing the dynamic content. After compilation, a JSP generates a servlets and therefore incorporates all the servlets functionalities.

* **J2EE Application:**

J2EE applications are complex access data from a variety of source and cater to a variety of client. To manage these applications the business function conducted in the middle tier. The J2EE platform acts as a middle tier and provides the necessary environment needed by the application. The J2EE platform provides” write once, run anywhere”, portability and scalability for multi-tier application. It also minimizes complexity for building multi-tier application.

To create a J2EE application we need to create following three components:

**(1) J2EE application client**

**(2) Enterprise Bean**

**(3) Web component**

Each of these components is packaged into a file with a specified file format. A J2EE application client is a Java application that run in a environment that enableit toaccess to the J2EE services. A J2EE application client is packaged into a .jar (Java archive) file. The web components are packaged into a .war (Web archive) file.

An Enterprise Bean consists of three files: the EJB class, Home and Remote Interfaces. The Enterprise Beans are bundled into an EJB.jar file. The .jar, .war and EJB.jar are assembled into a J2EE application, which is an .ear file. The .ear file is then deployed to the J2EE server.

The race for market share in the database industry has increased with the advent of client-server platforms. Oracle is one of the most successful companies that has released a number of development tools including SQL \*PLUS, PL/SQL that enables faster and easier application development and its management.

Oracle is the robust Database System, it support very large database. Moreover Oracle is widely used as back end for client / server applications. Administrative tools of Oracle help in securing the Data / Information.

**Process of creating a J2EE application:**

Component

(.jar file)

Enterprise Bean

(.jar file)

Assembled

J2EE Application Server

(.jar file)

Deployed

J2EE Server

* **J2EE Technologies:**

# The J2EE includes many technologies such as:

* Enterprise Java Beans (EJB)
* Remote Method Invocation (RMI)
* Java Naming and Directory Interface (JNDI)
* Java Database Connectivity (JDBC)
* Java Transaction API (JTA)
* Java Transaction Services (JTS)
* Java Messaging Services (JMS)
* Java Servlet & Java Server Pages (JSP)
* Extensible Markup Language (XML)
* **EJB:**

Enterprise Java Beans (EJB) is “write once, run anywhere” middle tier component consisting of method that implements the business rule. Enterprise Bean encapsulates the business logic. There are two types of Enterprise Bean: Entity Bean and Session Bean.

* **RMI:**

Remote Method Invocation is defined for the communication of remote objects in the middle tier of the distribute application. It enables a Java object to communicate remotely with other Java object.

* **JNDI:**

Java Naming and Directory Interface is an extension to Java platform and provide multiple Naming and Directory services. A Naming services provide a mechanism for locating distributed object. A Directory services organize the distributed object and other resources such as file in hierarchical structure. Directory services allow resources to be linked virtually so as located in to directory services hierarchy. There are different types of Directory services. JNDI allows the different types of Directory services to be link. Thus client can use any type of directory services.

* **JDBC:**

Java Database Connectivity provides a Database programming API for Java program. A JDBC API contains a set of classes and Interfaces that are used to connect a database build using any DBMS or RDBMS. It also submit SQL query to a database and retrieve its and processes the result of SQL query.

* **JTA & JTS:**

Java Transaction API (JTA) and Java Transaction Service (JTS) are transaction API. One can use these API to democrat whether the transaction starts or ends.

* **JMS:**

Java Messaging Service is an API that J2EE platform include to send mail via Internet.

* **Servlet:**

Servlets are used to develop a variety of web-based application. They make use of the extensive power of the Java API such as networking and URL access, multithreading, database connectivity, internationalization, RMI and object serialization. Java Server Pages (JSP) adds server side programming functionality to Java. Both Servlet and JSP allow the creation of database driven web application and have server side programming capability.

* **XML:**

J2EE uses Extensible Markup Language as a Markup language to describe the contents. The described file created when deploying the J2EE application is an XML file.

* **J2EE SDK TOOLS:**

J2EE SDK includes following tools:

1. The Deployment Tool.
2. The J2EE Server.
3. The Cloud Scale Server.
4. The Clean-up Script.
5. The Packager Tool.
6. The Realm Tool.
7. The Run Client Script.
8. The Verifier Tool.

* **The J2EE Security:**

The architecture of the J2EE is such that it enforces security in the application. In order to access the J2EE services, a user need to prove his/her identity. Such users are called J2EE users and process is called authentication. The J2EE authentication services are different from security of the operating system. The users of the operating system and the users of the J2EE belong to a different realm. A realm is a group of users that have the same authentication policy. The users of J2EE belong to a two different realms that are respectively authentication by certificates and defaults. J2EE certificate to authenticate a web browser client. In most cases, the J2EE services use the default realm to authenticate a user. J2EE users may also belong to a group. A group is a collection of users who have common feature for eg. The user belonging to a group may all belonging to a group coding same module. Similarly project managers might belong to a different group.

When J2EE application client execute its request that you enter login id and password. If the combination of both username and password correct the J2EE allow you to access the services.The J2EE server also enforces security by process known as authentication. Authorization is a process by which the permissions are assigned by server to invoke the method of Enterprise Bean.

JAVA BEANS

The java beans specification allows software components to be written in java, which encapsulate the logic behind the web application and remove the bulk of the code that would otherwise clutter up JSP’s. The result is JSP code that is simpler, easier to maintain and which is more readily accessible to non-programmers.

A bean uses properties to describe internal data that affects how it works and what it shows. In java the actual bean property data is usually a private or protected field, which can be edited by publicly available methods. In other words beans allow access to internal data via public get and set methods. This confirms to object orientation norms, which hide internal data from users and explore it only through accessor methods. Another aspect of this component is that it should be able to communicate with other objects or beans. Java beans accomplish this by firing events and listening to them. A bean that is interested in what happens to an object external to itself can register itself as a listener for various events in that object. Conversely an external object can register itself to listen to that bean. This concept is really the key to providing standalone software components.

**BOUND PROPERTIES**:

One way of exporting events is to use bound properties. When a property value changes a bound property can inform other parts of the application that its value is changed.

**BEAN EVENTS***:*

The property change support class should cater for most of the needs. However on many occasions beans will still need to communicate even though no property change activity has occurred.

**BEAN PERSISTANCE AND STORAGE**:

For a component to be really useful it must be possible to save it and any values it may contain and reload it to the same state at a later date. No matter how wonderful the component we create it will not be used much if it has to reset every time it is retrieved. Java beans use the serializable interface to address these issues.

The serializable interface has no methods to implement. It is simply on indicator to the compiler that object may be made persistence by serialization. In practice serialization generally means saving the bean to a file using the object output stream classes. Then to restore the bean to read from the same file using object input stream. To make java bean to be serializable we need to do is add the serializable interface in the class declaration.

JDBC

There are many classifications of databases available as Hierarchical database, Network database, Relational database, Object databases and soon. Due their flexibility Relational database management systems are most successful bread of databases in the history of computing. Ex: - Oracle, IBMdb2, and Microsoft SQL Server.

A technology that enables JSP base applications to interact directly with database engines is called Java Database Connectivity and is an integral part of Java platform. JDBC/JSP based web application access the database connections. These connections must be managed carefully by the application especially if a large number of concurrent users may be accessing them. To make this performance optimization JDBC uses a mechanism called connection pooling. The evaluation of this open database access technology has led to a mirade of driver architecture.

**Interaction of JSP Page with JDBC**

Application Server Machine

Client Machine

**Browser**

**with**

**HTML**

DB Server Machine

Here the browser using the web application is not required to support java at all. The JSP has full control over how many JDBC connections are made to the server. The client never makes direct JDBC connection to the server. This solution can work readily through a firewall, only standard HTTP is used between the web server and the client.

As a bonus this solution sends itself to easily secured information simply by adding secured socket layer support to the web server. Because of this separation of the presentation from the business logic, which is separated from the database logic, this sort of system is often called three tiers of the system. Although the application server and database server can also running on the same server machine.

There is still one minor problem with this scenario. Project personal accessing the JSP page containing the embedded JDBC code can easily and inadvertently modify the database access code and this may result in an erroneous application or even corrupted database. There are 2 solutions for this:

1. Create java beans or java classes that encapsulate all the JDBC operations. This is significantly better solution. But instantiation, initialization and parameterization of the java class or the beans can still represent a significant amount of embedded java code with in the JSP.
2. Create a tag extension set to ‘pushdown’ all the database access logic. The data access logic programmers write the set of custom tags. The JSP application logic designers will then use the set of custom tag to create their application.

### JSP Engine

|  |
| --- |
| JSP’s  Custom tag Custom Java  extension beans/classes  JDBC Driver Other DB access Technologies |

**BECK-END**

**ORACLE 8i**

**Why we are using Oracle (RDBMS)?**

Some of the merits of using Oracle (RDBMS) are as under:

* Centralization of database.
* Client Server Technology.
* Security.
* Normalization of Data Base.
* Relationship.
* Transaction Processor.
* It gives some internet related features.

Hence because of these features we are using Oracle as a back-end technology.

Weather you are working on LAN projects or Distributed projects, there are two sides of it:-

* Front End
* Back End

Front End remains on client side. Front end is made for end user who uses our application. Basically in front end, our input-output forms reside which takes the input from the client and gives output back to client.

Backend remains on server side and has two components viz.

* Server side programs
* Data Base

Database is the most important thing in this universe as database gives identity to a thing. It provides us with a repository where we can store ample amount of data, at one place. Without a database, existence of a thing is impossible.

While working on a project first step is to design a database.

**What is a database?**

Data Base is a collection of tables and table is a collection of records in a tabular form i.e. in row and columns format.

Data Base can be divided into two parts:-

* RDBMS
* DBMS

We will be using RDBMS (Relational Database Management System) in our project i.e. oracle 8.0 Enterprise Edition.

**ABOUT ORACLE 8.0**

Oracle 8.0 contains all the features of previous version. It also supports some new features & enhancement to some existing features. Oracle servers provide deficient & effective solution for the major features.

* **Large Database & Space Management Control**

Oracle supports the largest database potential of hundreds of Giga Bytes in size. To make efficient use of expensive devices, it allows full control of space usage.

* **Many Concurrent Database Performances**

It supports large no of concurrent users executing a variety of database applications operation on the same data. It minimizes data connection & guarantees data concurrency.

* **High Transaction Processing Performance**

Oracle maintains the processing features with a high degree of overall system performance. Database users don’t suffer from slow processing performance.

* **High Availability**

Oracle works 24 hours a day with no downtime or limited database throughput. Normal system operation such as database backup & partial system failure doesn’t interrupt database use.

* **Controlled Availbility**

Oracle can selectively control the availability of data at the database level & sub database level. E.g. an administrator can disallow use of a specific application .Data can be reloaded without affecting other application.

* **Industry Accepted Standards**

Oracle adheres to industry accepted standards for the data access language operating system, user interface & network communication protocols.

* **Manageable Security**

To protect against unauthorized database aspects & users, Oracle provides failsafe security features to limit & monitor the data area. The system makes it easy to manage even the most completed designs for data assets.

* **Database Enforced Integrity**

Oracle enforces data integrity “Business rules”, that dictate the standards for applicable data. As a result, the cost of coding & managing checks in many database applications is eliminated.

* **Distributed Database System**

For community environment that are connected via networks, Oracle combines the data physically located on different computers in one logical database that can be accessed by all the network users. Distributed systems have same degree of user transparency & data consistency as non-distributed systems, yet receive the advantages of local database management.

* **Portability**

Oracle software is compatible to work under different operating system & same on all system. Applications developed on Oracle can be used on virtually any system with little or no more modification.

* **Compatibility**

Oracle software is compatible with industry standards, including most industry standard operating systems. Applications developed on Oracle can be used on virtually any system with little or no modification.

* **Connectivity**

Oracle software allows different types of computers & operating system to share information in networks.

# NEW FEATURES OF ORACLE 8.0

* **Improved Scalability**

The maximum size of an Oracle database has been increased to support hundreds of terabytes depending on the operating system on which it resides.

* **Improved Security**

Oracle 8.0 server now includes password management so that a password has a limited lifetime & must meet certain complexity such as minimum length. An account can be locked after a specified number of failed login attempts.

* **Improved Performance via Partition**

A table of index can be divided into smaller pieces called partitions, based on the value of one or more columns. A table partitions can be individually managed so that operation in one partition does not affect the availability of data on other partitions. Also insert, update, delete operations against a partitioned table can be processed partially.

In other words, the Oracle 8 server can assign a portion of the work to execute a single DML statement to multiple processes, which may then be allocated to multiple processes by the server operating system. As a result, the parallel DML operations are completed more quickly.

* **Enhanced Support for Database Replication**

The performance & manageability of database replication has been significantly improved.

* **Capability to handle a much larger number of concurrent users**

By pooling database connection, the Oracle 8 server is able to service a much larger number of concurrent users, up to 3000, depending on the server’s operating system & server hardware resources.

* **New & Improved Data Types**

Some existing data types have been enhanced & new data types have been introduced.

* **Improved Select Statement**

A new feature of the select statement allows a sub query to be used in place of a table in a from clause.

Now when we are discussing Database, there is one more thing attached to it, i.e. “Data Base Models”

**Database Models**

There are three kinds of database models:-

* **Single tier architecture.**
* **Two tier architecture.**
* **N- Tier architecture**.

**Single tier Architecture:**

In this kind of architecture, database and client application remains on one machine i.e. there is no client-server technology, there is no centralization of database, and basically it is a stand alone system.

**Two tier Architecture**

In this kind of architecture, database and client application is on two different machines i.e. Database on one machine and the application on another machine. In this type of architecture, the implementation of client-server technology is done and centralization of data base is there, but it has two demerits:-

* Security is not there
* Multiple Client access is not there.

**N- Tier Architecture:** - In this kind of architecture, there is a middle-ware in between the client and database. Middle ware checks the validity of the client i.e. weather the client can access the database or not. Hence there is security in it as well as middle-ware allows multiple clients access.

**What is Middle-Ware?**

Middle-ware is a concept. Middle-ware provides centralization of business logic i.e. instead of putting logic on each and every client machine we put logic on a centralized server. Hence middle ware is nothing but a server side program where all your business logic and business methods reside. It remains on server side and it has all the logical building. Middle ware provides: -

* Multiple Client access.
* Centralized business logic in case of distributed application.

Because we are working on Distributed Application Based Project we need platform independent Language like **Java**

SOFTWARE

AND

HARDWARE TOOLS

Software and Hardware Tools

**Development Environment:**

* **Operating System: Windows XP**

The system will be built on windows compatible environment. The application will be web based developed using Java technology.

* **Web Server:**

**BEA’s WebLogic 8.1** Application Server to serve as Servlet/JSP engine.The system requires WebLogic Application Server for serving the requests for Servlet.

* **Server side Application Software**: Java Server Pages (JSP)
* **Business Logic Software**: Java Beans. (JB)
* **Client Side Application Software**: Java Script, HTML
* **Data Base**: Oracle 8i

The system requires Oracle as a database; however the system will be ODBC complaint to work on any standard database.

* **Client Browsers:**

Internet Explorer 5.0 or Netscape Navigator 4.7

The system requires Internet Explorer or Netscape Navigator browser for client side.

* **Dream Weaver 8.0**

The system will be developed with Java Technologies using J2SE (JDK and JRE). Dream Weaver 8.0/ Front Page 2003 as HTML editor.

**Hardware & Software Requirements**

**Hardware requirement**

Main Processor Pentium IV

Hard-disk Capaity 8 G.B

RAM 256 MB

Clock Speed 2.8 Hz

Floppy Drive : 1.44MB

Keyboard 104 Key

Monitor V.G.A

4.2 Software specification

**Software Requirement**

Operating System Window XP

Backend tool Oracle 8i

Front-end tool Java