1 Introduction

1.1 Problem Statement:

This project is aimed at developing an online application for the Training and Placement Dept. of the college. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an application for the TPO of the college to manage the student information with regards to placement. Students logging should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Students.

1.2 Purpose:

For the purpose of training and placement of the student in colleges, TPO’s have to collect the information and CV’s of students and manages them manually and arranges them according to various streams.

If any modification is required that is to be also done manually. So, to reduce the job required to manage CV’s and the information of various recruiters, a new system is proposed which is processed through computers.

1.3 Document Conventions:

Font-Style : Times New Roman

Font-Size:

Main Headings: 16

Sub Headings: 14

Content: 12

Line Spacing: 1.5

1.4 Intended Audience:

1.5 Scope of Project:

Our project has a big scope to do. We can store information of all the students. CV’s are categorized according to various streams. Various companies can access the information. Students can maintain their information and can update it. Notifications are sent to students about the companies. Students can access previous information about placement.

1.6 References:

www.

SYSTEM STUDY AND ANALYSIS

2.1 Product Information:

2.1.1 Product Perspective:

This project is mainly intended for automating this procedure that can help the people who belong to the T&P cell by saving their time ,based on this basic operation actually their activity is under two steps the first one is we have to maintain the list of students and their credit records and the second job is to maintain the company details and based on the company requirements we need to select the students and we need to make the list of students branch wise, which is more complex task, and here informing is through notice boards , where as this is also a bit old fashioned task, which can be automated in our proposed system by sending mails to the respective candidates. This proposed system is far advantageous than the existing one in many cases such as retrieving the student details is easily maintained in a manner that with just one click we can easily attain the details of the company such as the responsible person contacts and company contact details such as address and phone numbers can be maintained.

T&P cell mainly include the details of students, based on these requirements specified by various companies, appropriate students list must be generated. The percentage of the students must be appropriate and true. We should be able to generate a notice so that we notify all the departments the corresponding information about campus recruitment drives. We need top view the student details.

**EXISTING SYSTEM:**

* Filling of forms by students

Here a form is given to students in which he/she has to fill with some details such as his name, roll number, contact details, percentages(from first year to till date), Intermediate particulars(name of institution, place, year of pass, percentage), SSC particulars(name of institution, place, year of pass, percentage).

* Collecting marks from green book

Marks of each student are collected from green book (a book containing marks and other details maintained by each department in college).

* Preparing excel sheet

From the data collected through filled-forms and green book, excel sheets are prepared.

These excel sheets are used to prepare a list of students who full-fill the requirements of a company visiting the campus and these students are eligible to attend the campus placement.

**PROPOSED SYSTEM:**

In the proposed system the user need not do all the hectic work. he will be provided with an interface with which he can easily get his work done.

The following are the facilities that are provided by the system to the user.

* Notice generation

Here user has to provide information to the system about company name, date and venue at which campus drive might take place. With this information the system will generate a notice (for example, a word document) which can be placed in college notice board to intimate students about placement drive.

* Student list generation

Here the user has to provide information to the system about the requirements of the company (such as, cut off percentage, number of backlogs allowed etc.) so that it will generate him a list of students who are eligible as per the requirement.

* View student profile

Here the user is able to view a student’s profile of his interest by giving the student’s roll number as input.

* Mailing

Here the user is provided to mail students or others (for example, company officials) depending on his need.

* Result analysis

Here the user is able to get the results which are released and store them for later usage.

.

**2.1.2 PRODUCT FEATURES**

* Complete automation is possible in this sector, which is against the main disadvantage namely time consuming.
* Can maintain student details who have been studying in the college.
* Any kind of lists based on students profile can be retrieved with in less time.
* Results are uploaded directly from net so that no errors exist in calculating percentages.
* Effective and good means of communication can be facilitated as we have included mailing module in the proposed system.

2.1.3 User Classes:

2.2 **Requirement Analysis**

We are overcoming the difficulty of student details which were manual in the current system and here we generate detailed information about the students which will save our time to inform each and every batch and section and student profile is maintained.

**2.2.1 Functional Requirements:**

This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. A faculty member should be able to login to the system through the first page of the application, and mention his required roll number and he should get the details of the student with that roll number. An administrator can login into his account and he will update the student information.

**2.2.2 Non Functional Requirements:**

**Usability**

This section includes all of those requirements that effect usability.

* We get the response within seconds.
* The software must have a simple, user-friendly interface so customers can save time and confusion.

**Reliability**

* The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

**Supportability**

* The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform which is having JVM built into the system. This application is being developed using J2EE; hence it is extremely portable.

**Implementation**

* The system is implemented in web environment. The apache tomcat is used as the web server and windows Xp professional is used as the platform.

**Interface**

* The user interface is based on the web browser. The application is developed using JSP and HTML along with DHTML.

## The Interface design is aimed at a flexible front-end communication to provide the user with clear information in navigating a user-friendly interface is planned.

2.2.3 Performance Requirements:

* The completely separate business logic at server side from the student interface ensures good performance.
* The system exhibits high performance because it is well optimized. The business logic is clearly separate from the UI.
* We get the response within seconds.

2.2.4 Data Requirements:

2.2.5 Hardware Requirements:

Processor : Intel p2 or later

RAM : 512 MB or More

Hard Disk : 40 GB or more

SYSTEM DESIGN

3.1 System Architecture:

3.2 Detailed Design:

3.2.1 Usecase Diagram:

3.2.2 Class Diagram:

3.2.3 Collaboration Diagram:

3.2.4 Sequence Diagram:

3.2.5 Activity Diagram:

3.2.6 State Machine Diagram:

3.2.7 Component Diagram:

3.3 Data Modelling:

3.3.1 ER Modelling: The schemas for the database application can be displayed by means of graphical notation known as Entity Relationship diagram.

The ER model describes data as entities, relationships and attributes.

ENTITIES AND ATTRIBUTES

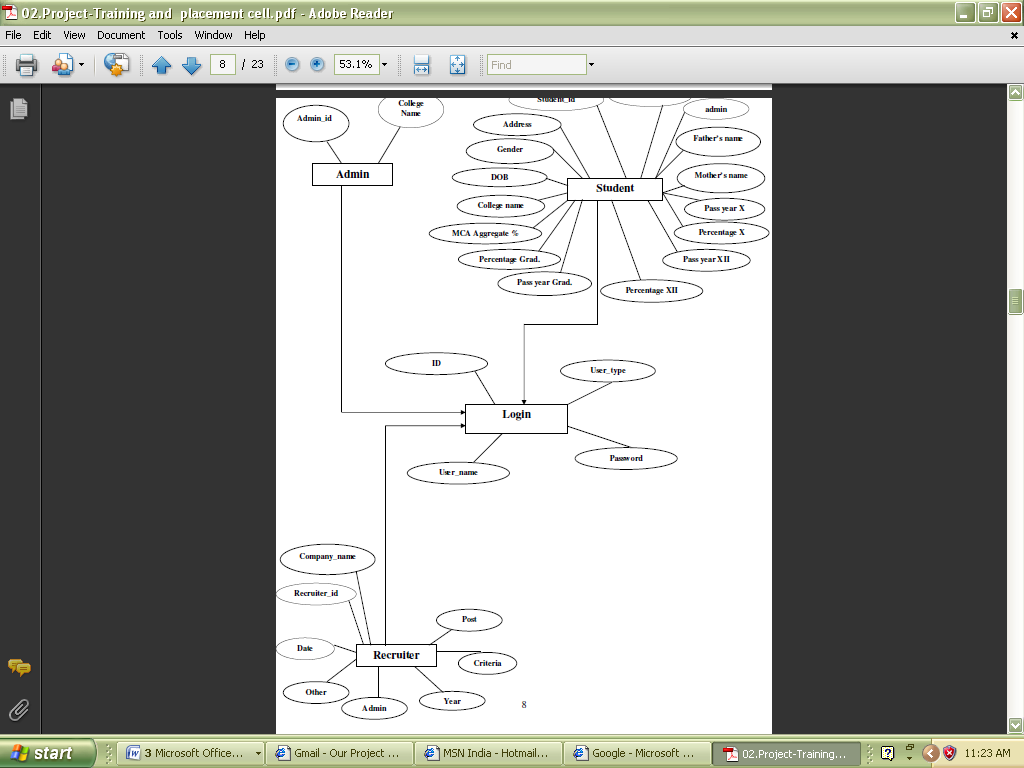
An entity may be an object with a physical existence (for e.g. A particular person, car or employee) or it may be an object with a conceptual existence (for e.g. a company, a job, or a university course)

Each entity has attributes i.e. the particular properties that describe it. The attribute values that describe each entity become a major part of the data store in the database.

RELATIONSHIPS BETWEEN ENTITIES

Whenever an attribute of one entity type refers to another entity type, a relationship exists. In the initial design of entity types, relationships are typically captured in the form of attributes. As the design is refined these attributes get converted into relationships between entity types.

In the ER diagrams the emphasis is on representing the schemas rather than the instances. This is more useful in the database design because a database schema changes rarely, where as contents of the entity sets change frequently.



3.3.2 Normalised Schemas:

3.1 User Interface Design:

4.Technologies:

**DESCRIPTION OF FRONT END**

**4.1.0 HYPER TEXT MARKUP LANGUAGE:**

The Hyper Text Markup language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specifications defines HTML version 4.0 HTML 4.0 aims to capture recommended practice as of early ’96 and as such to be used as a replacement for HTML 3.2

**Why to use HTML:**

Web site is a collection of pages, publications and documentation that reside on web server. While these page publication and a document as a formatted in any single format you should use HTML for home page and all primary pages and the site. This will enable the millions of web users it considered first formatting any new material you plan to publish on the web HTML documents are platform independent, meaning that they don’t conform to any standard it they are created properly you can more home page to any server platform or you can access them with any complaint www browser.

1. <HTML>…</HTML> - All HTML files start and end with the tag pair.
2. <HEAD>…</HEAD> - All HTML have a pair of “HEAD” tags that indicate what the tile and other attributes of the page are going to be.
3. <TITLE>…</TITLE> - this tag indicates what the title of the HTML file is going to be on the BROWSER window title.
4. <BODY>…</BODY> - this tag pair is to logically separate the HTML file into the header and the body. Usually the header contains information regarding the html where as the body contains information that the HTML file must actually contain.
5. The HTML template must look like.

<! DOCTYPE HTML PUBLIC “THIS IS AN EXAMPLE”>

<HTML>

<HEAD>

<TITLE> YOUR TITLE GOES HERE</TITLE>

</HEAD>

</HTML>

1. <P>…</P> - This tag pair used to indicate the paragraph. Any text that needs to be separated into a paragraph must be put in within a paragraph tag.
2. <B>…</B> - This tag pair is used to indicate the text within tag pair must be in bold letters.
3. <I>…</I> - This tag pair is used to indicate the text within the tag pari must be in italic letters.
4. <IMG SRC=”../images/corp.gif” ALT=:LOGO” HEIGHT=”100” WIDTH=”100”> - This tag is used to embed images in the HTML pages. The SRC attribute is used to locate the file name under a directory, the ALT attribute is used to indicate the TOOLTIP message that must appear, and HEIGHT and WIDTH indicate the height and the width of the images that is being shown on the HTML pages.
5. <H1 ALIGN=”CENTER”>…</H1> - This pair of tags is used to indicate that the text must be main title for the HTML page. The ALIGN attribute can be used to set the alignment to “center” or “left” or “right”
6. <H1>Heading1</H1>

<H2>Heading2</H2>

<H3>Heading3</H3> - This set of tags will show the Headings in smaller fonts as the heading increases.

1. ALIGN – The align attribute can be used for headings as well. For <P>…</P> tags also, the ALIGN attribute can be used.
2. <BR> - Used to insert a carriage return in the HTML file. The attribute to be used for this is the CLEAR attribute.
3. <CENTER>…</CENTER> - To center the entire block of text this tags are used.
4. <PRE>…</PRE> - These HTML tags are used when you want to preserve the formatting in a HTML block of text. The formatting could mean spaces, return and alignment etc.
5. <MARQUEE> - This is a MS-IE specific tag.
6. <A>…</A> - Anchor Tags. These tags are used linking namely hyper linking.

**Example:**

<A HREF=<http://www.ibm.com>>Visit IBM Web Pages</A>

1. Images Basics: Image Tag is used to embed images in the html document. The general syntax is

<IMG SRC=”logo.gif”>

1. Tables <TABLE>…</TABLE> - This is used to specify the table type of layout in the HTML document.

<TABLE BORDER=”1”>

<CAPTION> This is the CAPTION of the TABLE</CAPTION>

<TR>

<TH>Car</TH>

<TH>Company</TH>

</TR>

<TR>

<TH>Concorde</TH>

<TH>Chrysler</TH>

</TR>

</TABLE>

1. Fonts: The <FONT> …</FONT> tag is used to specify text in a particular font.

**Example:**

<FONT SIZE=”10”> this is a line of text with size 10</FONT>

Syntax:<FONT>…</FONT>

**4.1.1 FEATURES OF JAVA LANGUAGE:**

**Java:**

The Java Platform consists of the Java Application programming interfaces (APIs) and the Java virtual machine (JVM). Java API’s are libraries of compiled code that you can use in your programs. They let you add ready-made and customizable functionally to save you programming time. The console printing capability is provided in the API ready for to use; you supply the text to print.

**Object:**

As the name object-oriented implies, objects are key to understanding object-oriented technology. You can look around you now and see many examples of real world objects: your dog you desk and your television set your bicycle.

Phases of life of an object:

1. Creating Objects
2. Using Objects
3. Cleaning up Unused Objects

**Creating Objects:**

Rectangle rect=new Rectangle();

This single statement performs three actions:

**Declaration:** Rectangle rect is a variable declaration that declares to the compiler that the name rect will be used to refer to Rectangle Object. Notice that a class name is used as the variable;s type.

**Instantiation:** new is a java operator that creates the new object (allocates space for it).

**Initialization:** Rectangle() is to call Rectangle’s constructor, which initializes the object.

**Definition:** an object is a software bundle of variable and related methods.

Class:

In the real word, you often have many objects of the same kind. For example, your bicycle is just one of the many bicycles in the world. Using object-oriented terminology, we say that you bicycle object is an instance of the class of object known as bicycles. Bicycles have some state (current gear, current cadence, two wheels) and behavior ( change gears, brake) in common. However, each bicycle’s state is independent of and can be different from other bicycles. In object-oriented software, it’s also possible to have many objects of the same kind that share characteristics: rectangles, employee records, video clips and so on. Like the bicycle manufacturers, you can take advantage of the fact that objects of the same kind are similar and you can create a blueprint for the objects. Software “Blueprints” for objects are called classes.

**Definition:**

A class is a blueprint or prototype that defines the variables and methods common to all objects of a certain kind.

**Constructors:**

Classes have a special method called a constructor that is called when a class instance is created. The class constructor always has the same name as the class and no return type. If you do not write your own constructor, the compiler adds an empty constructor, which calls the no-arguments constructor of its parent class. The empty constructor is called the default constructor. The default constructor initializes all non-initialized fields and variable to zero.

**Abstract class:**

Programmers can implement super classes called abstract classes that define “generic” behaviors. The abstract super class defines and many partially implement the behavior but much of the class is undefined and unimplemented. Other programmers fill in the details with specialized subclasses.

In object-oriented programming, you may want to model an abstract concept without being able to create an instance of it. For example, the number class in the java. Lang package represents the abstract concept of numbers. It makes sense to model numbers in a program, but it doesn’t make sense to create a generic number object. Instead, the number class makes sense only as a super class to classes like integer and float, both of which implement specific kinds of numbers. A class such as number, which represents an abstract concept and should not be instantiated, is called – it cannot be instantiated. To declare that your class is an abstract class, use the keyword abstract before the class keyword in your class declaration:

Abstract class Number

{

….

}

**Interface:**

A java interface defines a set of methods but does not implement them. A class that implements the interface agrees to implement all of the methods defined in the interface, there by agreeing to certain behavior.

**Definition:**

An interface is a named collection of method definitions (without implementations). An interface can also include constant declarations.

**Package:**

A package is collection of related classes and interfaces that provides access protection and namespace management.

You should bundle these classes and the interface together in a package, for several reasons.

1. You and other programmers can easily determine that these classes and interfaces are related.
2. You and other programmers know where to find classes and interfaces that provide graphics related functions.
3. The names of your classes won’t conflict with class names in other packages because the package creates a new namespace.
4. You can allow classes within the package to have unrestricted access to each other, yet still restrict access for classes outside the package.

To create a package, you simply put a class or interface in it. To do this, you put a package statement at the top of the source file in which the class or interface is defined.

For example, the following code appears in the source file Circle.java and puts the Circle class in the graphics package:

Public class Circle extends Graphic implements Drag gable

{

…

}

The Circle class is a public member of the graphics package.

**4.1.2 JAVA 2 ENTERPRISE EDITION:**

Technically J2EE is not a language; it is a group of specifications, frameworks, technologies, etc. for building distributed enterprise systems. J2EE is comprised of a number of programming and scripting languages including Java, XML, JSP, HTML, SQL and others.

Some of the advantages of J2EE include cross-problem portability, availability of open-source libraries, a huge server-side deployment base, and coverage for most W3C standards.

**4.1.2(a) SERVLETS:**

Servlets are modules that extend request / response oriented servers, such as java-enables web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business login used to update a company’s order database. Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which you use to write servlets, assumes nothing about servers environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support servlet API. Use servlets instead of CGI scripts.

**Other uses of Servlets:**

Here are a few more of the many applications for servlets. Allowing collaboration between people.

A servlet can handle multiple requests concurrently and can synchronize requests. This allows servlets to support systems such as on-line conferencing.

Forwarding requests:

Servlets can forward request to other servers and servlets. Thus servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service other several servers, according to task type or organizational boundaries.

**The Servlet Interface:**

The central abstraction in the servlet API is the servlet interface. All servlet simple implement this interface, either directly or more commonly, by extending the class that implements it such as HttpServlet.

The servlet interface declares but does not implement methods that manage the servlet and its communications with clients. Servlet writers provide some or all these methods when developing a servlet.

**Client Interaction:**

When a servlet accepts a call from a client, it receives two objects:

* Servlet Request
* Servlet Response.

**A Servlet Request:**

This encapsulates the communications from the client to the server.

**A Servlet Response:**

Which encapsulates the communications from the servlet back to the client. Servlet Request and Servlet Response are interfaces defined by the javax.servlet package.

**Servlet Request Interface:**

The Servlet Request interface allows the servlet access to:

Information such as the names of parameters are passes in by the client, the protocol (Scheme) being used by the client, and the names of the remote host that made the request and the server that received it.

The input stream: servlet input stream: servlet use the input stream to get data from clients that use applications protocols such as HTTP POST and PUT methods.

Interfaces that extend Servlet Request interface allow the servlet to retrieve more protocol-specific data. For example, the HttpServeltRequest interface contains methods for accessing HTTP specific header information.

**The Servlet Response Interface:**

The servlet response interface gives the servlet methods for replying to the client. It allows the servlet to set the content length and MIME type of the reply.

Provides an output stream: servlet output stream and the writer is that through which the servlet can send the reply data. Interfaces that extend the servletResponse interface give the Servlet more protocol specific capabilities. For example, the HttpServletResponse interface contains methods that allow the servlet to manipulate HTTP- specific header information.

The methods to which the service method delegates HTTP requests include, doGet, for handling GET, conditional GET, and HEAD requests.

doPost, for handling POST requests.

doPut, for handling PUT requests.

doDelete, for handling DELETE requests.

**Requests and Responses:**

Methods in HttpServlet class that handle client requests take two arguments: An HttpServletResponse object, which encapsulates the response to the client.

The Life Cycle of Servlet: each servelt has the same life cycle. A server loads and initializes the servlet. The servlet handle zero or more client requests. The server removes the servlet (some servers do this step only when they shut down)

**Advantages:**

OOP servers several benefits to the program designer and the user. Object orientation contributes to the solution of many problems associates with the development and the quality of software products. The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost. The principle advantages are through inheritance, we eliminate redundant code extend the user of existing objects.

**Disadvantages:**

The runtime cost of dynamic binding mechanism is the major advantage of object oriented languages. The following were demerits of adopting object orientation in software developments in the early days of computing (some remain forever)

1. Compiler overhead.
2. Runtime overhead
3. Reorientation of software developer to object oriented
4. Thinking
5. Requires the mastery over the following areas:
   1. Software engineering
   2. Programming methodologies.
6. Benefits only in long run while managing large software projects at least moderately large ones.

**4.1.2(b) JAVA SERVER PAGES:**

Java Server Pages (JSP) is a J2EE component that enables you develop web applications that work as if they had been created in java servlets. The JSP specification defines JSP as “a technology for building the applications for generating dynamic web content such as HTML, DHTML, SHTML and XML”.

H

T

M

L

Server

ServletEngine

JSP

Engine

If the request comes for a JSP file for the first time then the servlet engine will call the JSP engine. Now the JSP engine will convert the JSP file into an servlet and then it will compile it there by making it a class file. Then this class file will be executed by servlet engine.

**Properties of JSP:**

1. In JSP we will the write the java coding inside HTML tags. To achieve this there are different tags in JSP.
2. The no. of the lines of code in JSP will be less when compared to servlet.

JSP elements can be grouped according to the functions they perform.

* Declarations
* Expressions
* Directives
* Script lets
* Comments
* Actions
* Implicit JSP objects

**Declarations:**

Declarations do not generate the output to the user’s screen. They are used for declaring the class variables and methods and start with <%!. The variable declared in the following code is available only in that page.

Eg. <%! String name;%>

The code contained in declaration block will be located in the generated servlet outside existing method.

**Expressions:**

JSP Extensions start with <%= and can contain any java expression, which will be evaluated and its results inserted into the HTML pages right where the expression is located.

Eg:

<HTML>

<BODY>  
<%! Double salary=5000;%>

Your new salary is <%=salary\*1.2%>

**Directives:**

Directives do not generate screen output. They inform the JSP engine about the rules to be applied to JSP. The page directive start with <% @ page and will be applied during the servlet-generation process only to the current page. It’s used with attributes as import, extends, sessions, error page and the content type.

The include directive allows inclusion of any text files or codes from another JSP, at the time when the page is compiled into a servlet.

**Scriptlets:**

Scriptlets can contain any valid java code that will be included in the method JSP serice during the servlet code generation.

Eg. <% lname=”java”;%>

The plain text and the HTML tags should be places outside the scriptlets.

Even though the JSP syntax enables the insertion of java code fragments, variables and method declarations, you should try to minimize the amount of java code in the JSP body.

**Comments:**

JSP comments start with <%-- and end with --%>, and are not included in the output web pages. If you need to include the comments in the source of the output page, use the HTML comments.

**Implicit JSP Objects:**

JSP provide a number of predefined variable that give us access to these vital objects.

* **Request** – this variable points at HttpServeltRequest.
* **Response** – use this object to access the HttpServletResponse object.
* **Out** – this variable represents the JSP Writer class, which has the same functionality as the print writer class in servlets.
* **Session** – this variable represents an instance of the HttpSession object.
* **Exception** – this variable represents an instance of the uncaught Throwable object and contains error information. This variable is only available from the JSP error page that contains the directive is ErrorPage=true
* **Page** – this variable represents the instance of the JSP’s java class processing the current request.

**4.1.3 JAVA DATABASE CONNECTIVITY:**

**Establishing a Connection:**

The first thing you need to do is establish a connection with the DBMS you want to use. This involves following steps:

**Loading Drivers:**

Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC bridge driver, the following code will load it:

**Class.forName(“sun.jdbc.odbc.JdbcDriver”);**

Your driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverXYZ, you would load the driver with the following line of code:

**Class.forName (“jdbc.DriverXYZ”);**

You do not need to create an instance of a driver and register it with the DriverManager because calling Class.forName will do that for you automatically. If you were to create your own instance, you would be creating an unnecessarily duplicate, but it would do no harm. When you have loaded the driver, it is available for making a connection with a DBMS.

**Making the Connection:**

The second step in establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea;

Connection con=DriverManager.getConnection(url,”mylogin”,”mypassword”);

**Creating JDBC Statements:**

A statement object is what sends your SQL statement to the DBMS. You simply create a Statement object and then execute it, supplying the appropriate. Execute method when the SQL statement you want to send. For a SELECT Statement, the method to use is executeQuery. For statements that create or modify tables, the method to use is ExecuteUpdate. It takes an instance of an active connection to create a statement object. In the following example, we use our Connection object to create the Statement object stmt:

**Statement stmp=con. createStatement();**

At this pint stmt exists, but it does not have an SQL statement to pass on to the DBMS. We need to supply that to the method we use to execute stmt. For example, in the following code fragment, we supply executeUpdate with the SQL stmt.executeUpdate(“CREATE TABLE COFFEES”+

”COF\_NAME,SUP\_ID INTEGER,PRICE FLOAT,”+

”SALES INTEGER, TOTAL INTEGER”);

Since we made a string out of the SQL statement and assigned it to the variable create TableCoffees, we could have written the code in this alternate form:

**Stmt.executeupdate(create TableCoffees);**

**Executing Statements:**

We used the method executeUpdate because the SQL statement contained in create TableCoffess is a DDL (Data Definition Language) statement. Statement that create a table, alter a table, or drop a table are all examples of DDL statements and are executed with the method executeUpdate. As you might expect from its name, the method executeUpdate is also used to execute SQL statements that update the table. In practice, executeUpdate is used far more often to update tables than it is to create them because a table is created once but may be updated many times.

The method uses most often for executing SQL statements is executeQuery. This method is used to execute SELECT statements, which comprise the vast majority of SQL statement. You will see how to use this method shortly.

**Using Prepared Statement:**

Sometimes it is more convenient or more efficient to use a prepared Statement object for sending SQL statements to the database. This special type of statement is derived from the more general class, statement that you already know.

**When to use a Prepared Statement Object:**

If you want to execute a statement object many times, it will normally reduce execution time to use a PreparedStatement object instead. The main feature of PreparedStatemet is that, unlike a statement object, is given a SQL statement when it is created. The advantage to this is that in most cases, this SQL statement when it is created. The advantage to this is that in most cases, this SQL statement will be send to the DBMS right way, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement’s SQL statement without having to compile it first.

Although PreparedStatement objects can be used for SQL statements, with no parameters, you will probably use them most often for SQL statements that take parameters. The advantage of using SQL statements that take parameters is that you can use the same statement and supply it with different values each time you execute it.

**4.1.3 JAVA SCRIPT:**

Java Script is new scripting language for web-pages. Script written with Java Script can be embedded into you HTML pages. With Java Script you have very many possibilities for enhancing you HTML pages with interesting elements. For example you are able to respond to user-initiated events quite easily. Some effects that are not possible with Java Script were some time ago only possible with CGI. So you can create really sophisticated pages with the help of Java Script. We can see many examples for Java Script on the internet.

**What is Different between Java and Java Script?**

Although the names are almost the same Java is not the same as Java Script. These are two different techniques for internet Programming. Java is a programming language (as the implies). The difference is that we can create real programs with Java. But often we just want you make a nice effect without having to bother about real programming. So Java Script is meant to be easy to understand and easy to use. Java Script authors should not have to extension to HTML than a separate computer language. Of course, this is not the official definition but I think this makes it easier to understand the difference between java and java script.

**How can Java Script Run?**

The first browser to support Java Script was the Netscape Navigator2.0 of course the higher versions do have Java Script as well you might know that Java does not run on all.

<HTML>  
 <HEAD>

MY First Java Script

</HEAD>

<BODY><BR>

This is normal HTML Document<BR>

<Script Language=”Java Script”>

Document.write(“this is Java Script”);

</Script>

Back in HTML again.

</BODY>

</HTML>

**4.2 Description of Back end**

**ORACLE 10g:**

The g stand for grid computing. A common misconception seems to be that grid is just the new name for RAC(having improved RAC). This is not the case. 10g comes with both RAC and grid. One will be able to install 10g with RAC only or with grid only, without either and with both. There is profound difference between grid and RAC

One of the goals for 10g was to deliver a complete, integrated stack of software so as to make it possible for oracle users to not have to depend any more on third party software.

This lead to 12 development focus areas for 10g:

* Application Development
* Business intelligence and data warehousing clustering
* Content management
* High availability
* Information integration
* Life sciences
* Location services
* Performance and scalability
* Security and directory
* Server manageability
* Windows

10g is said to have 149 new features. Possibly, the mose popular will be the modelclause and the transportable table spaces.

The SQL\*PLUS copy command will be deprecated.

* **SQL model clause**: This will enhance SQL for calculations. SQL result sets can be treated like multi dimensional arrays.
* **HTML DB**: This will be RAD environment for web based applications.
* PHP will be supported
* **SQL**: Regular expression (finally), native numbers (based on IEEE 754), enhancements for LOBs, enhancements for collections.

It should be noted, however, that regular expressions were available through the owa pattern package.

Data pump replaces EXP and IMP. It provides high speed, parallel, bulk data and meta data movement of oracle data base contents across platforms and data base versions. If a data pump job is started and fails for any reason before it has finished, it can be restarted at a later time.

* **ASM:**automatic storage management
* Flashback database: Old database block images are stored in a flash recovery area which allow fast rollbacks of database(as no online redo logs are required). Flashback database makes it also possible to correct user errors. undraping tables automatic shared memory management is another self management enhancement to oracle. It includes a new parameter: sga target
* **ADDM**: automatic database diagnostic monitor. ADDM enables oracle to diagnos its own performance problems.

For example, ADDM identifies the most resource intensive SQL statements and passes that statement to the SQL tuning advisor.

* **AWR**: Automatic work load repository. AWR periodically gathers and stores system activity and workload data which is then analysed by ADDM.

**4.2(a)ORACLE 10g EXPRESS EDITION:**

Oracle database 10g express edition is an entry-level, small-footprint database based on the racle database 10g release 2 code base that’s free to develop, deploy and distribute; fast to download and simple to administer. Oracle data base XE is a great starter database for:

* **Developers** working on PHP, java, .NET, XML, and open source applications.
* **DBAs** who need a free, starter database for training and development.
* **Independent software vendors(ISVs) and hardware vendors** who want a starter database to distribute free of change.
* **Educational institutions and students** who need a free database for their curriculum

With oracle database XE, you can now develop and deploy applications with a powerful, proven. Industry-leading infrastructure, and then upgrade when necessary without costly and complex migrations.