

**CONTRIVANCE PLANNING MANAGEMENT SYSTEM**

**D.VAISHNAVI  
12PCA017**

**A Project Report Submitted to  
Avinashilingam Institute for Home Science and Higher Education for Women,  
Coimbatore-641043**

**In Partial fulfillment of the Requirements for the  
Master's Degree in Computer Applications**

**March, 2015**

# **CONTRIVANCE PLANNING MANAGEMENT SYSTEM**

**D.VAISHNAVI**

**12PCA017**

**A Project Report Submitted to  
Avinashilingam Institute for Home Science and Higher Education for Women,  
Coimbatore-641043**

**In Partial fulfillment of the Requirements for the  
Master's Degree in Computer Applications**

**March, 2015**

**Signature of the Supervisor**

**Signature of the Head of the Department**

**Signature of the External Examiner**

## ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for his constant love and grace that he has showered upon me.

I am very grateful to **Dr.T.S.K.Meenakshi Sundaram, M.A., M.Phil., Ph.D., Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of my study.

I heartily thank **Dr. (Mrs). Sheela Ramachandran M.Sc., P.G. Dip., Ph.D., Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my humble gratitude to **Dr. (Tmt). A.Venmathi M.Sc., M.Phil., Ph.D., Registrar, In charge**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I am also thankful to **Dr. (Mrs) A.Parvathi M.Sc., Dip.Ed. M.Phil., Ph.D., Dean Faculty of Science**, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr. (Mrs).G.Padmavathi M.Sc., M.Phil., Ph.D., Professor and Head**, Department of Computer Science, for providing all the facilities to complete the project.

I owe great deal of gratitude to my esteemed guide **Dr. (Mrs).I.Elizabeth Shanthi M.Sc., M.Phil.,Ph.D., Associate Professor** Department of Computer Science, for imparting the tremendous assistance and well timed support for triumph of my project.

I take this unique opportunity to express my sincere thanks to my project Coordinator **(Mrs).R.Vijayabhanu MCA., M.Phil., Ph.D., Assistant Professor, Department of Computer Science**, for her kind advice and knowledgeable suggestion, which helped me to complete my project successfully.

I would like to express my sincere gratitude to all the staff members of the Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women University, for their constant encouragement and for the opportunity to do my project in this esteemed university.

I also wish to thank all other teaching and non teaching staffs who stood behind the screen in making the project.

Last but not least, I take pride to thank my parents and family members for their support encouragement and kind blessings and also to my well wishers, friends who have helped us during the course of the project work and have made it a great success.

## **SYNOPSIS**

The basic idea of this project is to develop a web based recruitment process for a concern and it is entitled as Contrivance Planning Management System.

Contrivance is a web based application where any candidate can register and do the search for the job. In this portal the companies can register and can post their requirement. The candidates (Job Seekers) can register and apply for the vacancies which are posted by the company and the client companies.

This application makes the searching process more efficient. It provides online information on job posting and job searching. The details of the client requirements candidate resumes, Employee details, and candidate's details will be maintained in the system which will reduce the loss of data and reduce the paper work. Hence this application will help the company to recruit qualified and eligible candidates for the concern.

# CONTENTS

---

---

S.NO	PARTICULARS	PAGE NO
1.	<b>INTRODUCTION</b>	
	<b>1.1 Problem Definition</b>	<b>1</b>
	<b>1.1 Overview of the Project</b>	<b>1</b>
2.	<b>SYSTEM CONFIGURATON</b>	
	<b>2.1 Hardware Specification</b>	<b>2</b>
	<b>2.2 Software Specification</b>	<b>2</b>
	<b>2.3 About the Software</b>	<b>2</b>
3.	<b>SYSTEM STUDY AND ANALYSIS</b>	
	<b>3.1 Existing System</b>	<b>25</b>
	<b>3.2 Proposed System</b>	<b>26</b>
	<b>3.3 Feasibility Study</b>	<b>26</b>
	<b>3.3.1 Technical Feasibility</b>	<b>27</b>
	<b>3.3.2 Economical Feasibility</b>	<b>27</b>
	<b>3.3.2 Operational Feasibility</b>	<b>27</b>
4.	<b>SYSTEM DESIGN</b>	
	<b>4.1 Input Design</b>	<b>28</b>
	<b>4.2 Output Design</b>	<b>29</b>

<b>5.</b>	<b>SYSTEM DEVELOPMENT</b>	
	<b>5.1 Modules</b>	<b>31</b>
	<b>5.2 Module Description</b>	<b>31</b>
	<b>5.2.1 Administration Module</b>	<b>31</b>
	<b>5.2.2 Employee Module</b>	<b>32</b>
	<b>5.2.3 Client Module</b>	<b>33</b>
	<b>5.2.4 Candidate Module</b>	<b>34</b>
<b>6.</b>	<b>SYSTEM TESTING ANDIMPLEMENTATION</b>	
	<b>6.1 System Testing</b>	<b>35</b>
	<b>6.2 System Implementation</b>	<b>43</b>
	<b>6.3 System Maintenance</b>	<b>44</b>
<b>7.</b>	<b>CONCLUSION</b>	<b>45</b>
<b>8.</b>	<b>SCOPE FOR FUTURE ENHANCEMENT</b>	<b>46</b>
<b>9.</b>	<b>BIBLIOGRAPY</b>	<b>47</b>
<b>10.</b>	<b>APPENDIX</b>	<b>48</b>
	<b>A) Flow Chart</b>	
	<b>B) Data Flow Diagram</b>	
	<b>C) Screenshots</b>	
	<b>D) Table Design</b>	

# **1. INTRODUCTION**

## **1.1 Problem Definition**

The main problem in the existing system is that the candidates and the clients are manually serviced and this process is time consuming. For making contact with the clients the consultancy person need to visit the particular client company and then get the details of the company requirement. This will take a lot of time and man power.

The problem here is that the clients' and candidates' details are maintained manually, which may be lost/damaged. Because of all these reasons the organization may lose its goodwill.

## **1.2 Overview of the Project**

The main objective of the project is that the client and the candidates can submit their details, requirement and the resumes online. And their information will be stored in the database of organization. This will reduce the loss and damage of any kind of data. This project will also help in consultancy to save time and get the client requirement through online.

This project will also help the candidates to search the job and will be helpful for the growth of the company to provide good service for their clients.

## **2. SYSTEM CONFIGURATION**

### **2.1 Hardware Requirements**

<b>Processor</b>	:	Intel Pentium IV
<b>Processor Speed</b>	:	250MHz to 667 MHz
<b>RAM</b>	:	128 MB to 512 MB
<b>Hard Disk</b>	:	4 GB or higher
<b>Keyboard</b>	:	Standard 104 enhanced keyboard
<b>Mouse</b>	:	Local PS/2

### **2.2 Software Requirements**

<b>Operating System</b>	:	Windows 98/2000/XP/NT
<b>Web Server</b>	:	Apache
<b>Web Browser</b>	:	Mozilla Firefox
<b>Front-End Tool</b>	:	HTML
<b>Client side Script</b>	:	JavaScript
<b>Server side Script</b>	:	PHP
<b>Back-End Tool</b>	:	My SQL
<b>Dynamic Tool</b>	:	CSS

### **2.3 About the Software**

#### **2.3.1 Introduction to PHP**

The software we selected for our project is PHP, which provides a frame for using different technology, like JavaScript, CSS. Reason for choosing PHP script is that, it's suitable for coding server side program and we can also add JavaScript in the source code for various

validations. PHP also provides some inbuilt tools for validating various fields in containing in a Form which is to be filled by the user.

Data base technology has been described as one of the most rapidly growing areas of computer and information science. It is a known exaggeration to say that many thousands of organizations have become critically dependent on the continued and successful operations of a database system. Organization needs security, scalability, easy to install the software and capable of storing huge amount of data so we selected MSQl as a backend.

PHP is an open-source page scripting/tinplating system that is very similar to JSP and ASP. It defines its own scripting language, which looks and feels a lot like Perl. JSP uses Java as its scripting language (although some implementations support JavaScript, such as Caucho). ASP uses VBScript.

PHP is very popular -- it is used on over a million web sites -- but its main advantage (IMHO) seems to be that the language, being more "script" and Perl-like, is less intimidating to the great unwashed mass of HTML monkeys and hackers. In the long run, JSP and Java provide a more powerful system.

## **Overview of PHP**

### **History**

The origins of PHP date back to 1995, when an independent software development contractor named Rasmus Lerdorf developed a Perl/CGI script that enabled him to know how many visitors were reading his online résumé. His script performed two tasks: logging visitor information, and displaying the count of visitors to the Web page. Because the Web as we know it today was still young at that time, tools such as these were nonexistent, and they prompted e-mails inquiring about Lerdorf's scripts. Lerdorf thus began giving away his toolset, dubbed *Personal Home Page* (PHP).

The clamor for the PHP toolset prompted Lerdorf to begin developing additions to PHP, one of which converted data entered in an HTML form into symbolic variables that allowed users to export them to other systems. To accomplish this, he opted to continue development in C code rather than Perl. Ongoing additions to the PHP toolset culminated in November 1997 with the release of PHP 2.0, or Personal Home Page — Form Interpreter (PHP-FI). As a result of PHP's rising popularity, the 2.0 release was accompanied by a number of enhancements and improvements from programmers worldwide.

The new PHP release was extremely popular, and a core team of developers soon joined Lerdorf. They kept the original concept of incorporating code directly alongside HTML and rewrote the parsing engine, giving birth to PHP 3.0. By the June 1998 release of version 3.0, over 50,000 users were using PHP to enhance their Web pages.

Development continued at a hectic pace over the next two years, with hundreds of functions being added and the user count growing in leaps and bounds. At the beginning of 1999, Net craft reported a conservative estimate of a user base surpassing 1,000,000, making PHP one of the most popular scripting languages in the world. Its popularity surpassed even the greatest expectations of the developers, as it soon became apparent that users intended to use PHP to power far larger applications than was originally anticipated. Two core developers, Zeev Suraski and Andi Gutmans, took the initiative to spearhead a complete rethinking of the way PHP operated, culminating in a rewriting of the PHP parser, dubbed the Zend scripting engine. The result of this work was seen in the release of PHP 4.

Taken directly from PHP's home, "PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly." This is generally a good definition of PHP. However, it does contain a lot of terms you may not be used to.

Another way to think of PHP is a powerful, behind the scenes scripting language that your visitors won't see! When someone visits your PHP webpage, your web server processes the PHP code.

It then sees which parts it needs to show to visitors (content and pictures) and hides the other stuff (file operations, math calculations, etc.) then translates your PHP into HTML. After the translation into HTML, it sends the webpage to your visitor's web browser.

It is also helpful to think of PHP in terms of what it can do for you. PHP will allow you to:

- Reduce the time to create large websites.
- Create a customized user experience for visitors based on information that you have gathered from them.
- Open up thousands of possibilities for online tools.
- Allow creation of shopping carts for e-commerce websites.
- HTML - Know the syntax and especially HTML Forms.
- Basic programming knowledge - This isn't required, but if you have any traditional programming experience it will make learning PHP a great deal easier.

In year 2000 there were only few hundred web sites developed in PHP. By year 2007 number increased to more than 2, 00, 000, 00 and number is increasing and increasing. Now a day's most popular web site with heavy traffic has been developed in PHP.

PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

Since PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems' Java Server Pages, and mod\_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symphony, Code Igniter, and Zend Framework, offering features similar to other web application frameworks.

The LAMP and WAMP architectures have become popular in the web industry as a way of deploying web applications. PHP is commonly used as the P in this bundle alongside Linux, Apache and MySQL, although the P may also refer to Python or Perl.

The National Vulnerability Database stores all vulnerabilities found in computer software. The overall proportion of PHP-related vulnerabilities on the database amounted to: 20% in 2004, 28% in 2005, 43% in 2006, 36% in 2007, and 35% in 2008. Most of these PHP-related vulnerabilities can be exploited remotely: they allow hackers to steal or destroy data from data sources linked to the web server (such as an SQL database), send spam or contribute to DOS attacks using malware, which itself can be installed on the vulnerable servers.

These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent (23 in 2008, about 1% of the total). Recognizing that programmers cannot be trusted, some languages include taint checking to detect automatically the lack of input validation which induces many issues. Such a feature is being developed for PHP, but its inclusion in a release has been rejected several times in the past.

Hosting PHP applications on a server requires a careful and constant attention to deal with these security risks. Its flexibility knows no end.

Installing PHP as a CGI binary rather than as an Apache module is the preferred method for added security.

## Syntax

```
<Html>
  <Head>
    <Title>PHP Test</title>
  </head>
  <Body>
    <? php echo'
Hello World
'; ?>
  </body>
</html>
```

PHP code embedded within HTML

PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not processed by PHP; however, non-PHP text is still subject to control structures described within PHP code. The most common delimiters are `<? Php` to open and `?>` to close PHP sections. `<script language="php">` and `</script>` delimiters are also available, as are the shortened forms `<? Or` or `<? =` (which is used to echo back a string or variable) and `?>` as well as ASP-style short forms `<%` or `<%=` and `%>`. While short delimiters are used, they make script files less portable as their purpose can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, `<? php` and `?>`, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is well-formed XML.

Both double-quoted (") and heredoc strings allow the ability to embed a variable's value into the string. PHP treats new lines as white space in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: /\* \*/ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text (e.g. to a web browser).

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. *If* conditions, *for* and *while* loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit signed integers. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is *NULL*. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, or heredoc syntax.

## Functions

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming, although it does support multiprocessor programming.

### 5.2 and earlier

Functions are not first-class functions and can only be referenced by their name, directly or dynamically by a variable containing the name of the function. User-defined functions can be created at any time without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero argument class constructor functions called with the PHP new operator, where parentheses are optional. PHP supports quasi-anonymous functions through the create function () function, although they are not true anonymous functions because anonymous functions are nameless, but functions can only be referenced by name, or indirectly through a variable \$function name (); in PHP.

### 5.3 and newer

PHP gained support for closures. True anonymous functions are supported using the following syntax:

```
Function get Adder ($x)
{
    Return function ($y) use ($x) {
        Return $x + $y;
    };} $adder = getAdder (8);
Echo $adder (2); // prints "10"
```

Here, get Adder () function creates a closure using parameter \$x (keyword "use" forces getting variable from context), which takes additional argument \$y and returns it to the caller. Such a function can be stored, given as the parameter to other functions, etc. For more details see Lambda functions and closures RFC.

## Speed optimization

As with any interpreted language, PHP scripts are stored as human-readable source code and are compiled on-the-fly by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the webpage is accessed, PHP scripts can also be stored in binary format using PHP compilers such as phc and road send.

Code optimizers aim to reduce the runtime of the compiled code by reducing its size and making other changes that can reduce the execution time with the goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the eAccelerator PHP extension. Another approach for reducing overhead for high load PHP servers is using an Opcode cache.

Opcode caches work by caching the compiled form of a PHP script (Opcode) in shared memory to avoid the overhead of parsing and compiling the code every time the script runs. An Opcode cache, APC, will be built into PHP 6.

### 2.3.2 MY SQL

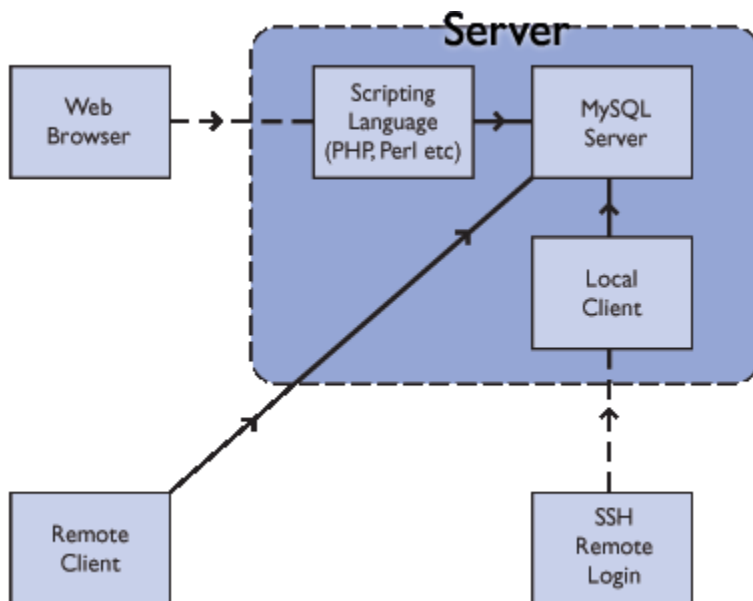
The Structured Query Language (SQL) is a very popular database language, and its standardization makes it quite easy to store, update and access data. One of the most powerful SQL servers out there is called MySQL and surprisingly enough, its free.

#### Some of the features of MySQL Include

- Handles large databases, in the area of 50,000,000+ records.

- No memory leaks. Tested with a commercial memory leakage detector (purify).
- A privilege and password system which is very flexible and secure, and which allows host-based verification.
- Passwords are secure since all password traffic when connecting to a server is encrypted.

MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements. SQL is a database language that is used to retrieve, insert, delete and update stored data. This is achieved by constructing conditional statements that conform to a specific syntax (i.e. the strict order required of elements for a statement to work). MySQL is a database server program and as such is installed on one machine, but can 'serve' the database to a variety of locations. To explain look at the following diagram.



The MySQL Server is installed on a Server and can be accessed **directly** via various client interfaces, which send SQL statements to the server and then display the results to a user. Some of these are:

**A Local Client** – is a program on the same machine as the server. An example of this is the command line MySQL client software we will be using in the rest of the MySQL workshops (although there are other programs including graphical interfaces).

**A Scripting Language** - can pass SQL queries to the server and display the result.

**A Remote Client** – is a program on a different machine that can connect to the server and run SQL statements. We can also use two more indirect methods.

**Remote Login** - We may be able to connect to the Server Machine to run one of its local clients.

**Web Browser** - We can use a web browser and scripts that someone has written (we're going to use this method for the rest of the workshop).

## Uses

Many web applications use MySQL as the database component of a LAMP software stack. Its popularity for use with web applications is closely tied to the popularity of PHP, which is often combined with MySQL. Several high-traffic web sites (including Flickr, Facebook, Wikipedia, Google).

## Platforms and interfaces

MySQL code uses C and C++. The SQL parser uses yacc and a home-brewed lexer, sql\_lex.cc MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, Open Solaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO Open Server, SCO exists

## Third party

Several other third-party commercial and non-commercial graphical administration applications (or "Front ends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known front ends are:

- **PhpMyAdmin** - a free Web-based front-end widely installed by Web hosts worldwide, since it is developed in PHP and only requires the LAMP stack to run.
- **HeidiSQL** - a full featured free front-end that runs on Windows, and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialized GUI features for date/time fields and enumerated multiple-value fields.
- **Navicat** - a series of commercial graphical database management applications, developed for Windows, Macintosh and Linux.

## Features

As of April 2009, MySQL offers MySQL 5.1 in two different variants: the MySQL Community Server and Enterprise Server. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- True Varchar support
- INFORMATION\_SCHEMA
- Strict mode

- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB, BDB and Cluster storage engines; save points with InnoDB
- SSL support
- Query caching
- Sub-Selects (i.e. nested Selects)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing (Index\_(database)) and searching using MyISAM engine
- Embedded database library
- Partial Unicode support (UTF-8 and UCS-2 encoded strings are limited to the BMP)
- Partial ACID compliance (full compliance only when using the non-default storage engines InnoDB, BDB and Cluster)
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via mysqlhotcopy) under certain condition.

The developers release monthly versions of the MySQL Enterprise Server. The sources can be obtained either from MySQL customer-only Enterprise site or from MySQL Bazaar repository, both under the GPL license. The MySQL Community Server is published on an unspecified schedule under the GPL and contains all bug fixes that were shipped with the last MySQL Enterprise Server release. Binaries are no longer provided by MySQL for every release of the Community Server.

## Distinguishing features

- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time):
  - Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Black hole, Cluster, Berkeley DB, EXAMPLE, and Maria)
  - Partner-developed storage engines (InnoDB, solids, NitroEDB, Info bright (formerly Bright house), Kick fire, XtraDB, IBM DB2)
  - Community-developed storage engines (memcache\_engine, http, PBXT, Revision Engine)
  - Custom storage engines
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

## MySQL storage engines

One powerful aspect of MySQL that sets it apart from nearly every other database server is that it offers users many choices and options depending upon the user's environment. From the server point of view, its default configuration can be changed to run well on a wide range of hardware. At the application development level, you have a variety of data types to choose from when creating tables to store records. But what's even more unusual is that you can choose the type of table in which the records will be stored. You can even mix and match tables of different types in the same database!

Storage engines used to be called table types. From time to time we refer to them as table types when it's less awkward to do so. In this chapter, we'll show the major differences between the storage engines and why those differences are important. We'll begin with a look at locking and concurrency as well as transactions—two concepts that are critical to understanding some of

Before drilling down into the details, there are a few general concepts we need to cover because they apply across all the storage engines. Some aren't even specific to MySQL at all; they're classic computer science problems that just happen to occur frequently in the world of multi-user database servers.

## MySQL Architecture

It will greatly aid your thinking about storage engines and the capabilities they bring to MySQL if you have a good mental picture of where they fit. Figure 2-1 provides a logical view of MySQL. It doesn't necessarily reflect the low-level implementation, which is bound to be more complicated and less clear cut. However, it does serve as a guide that will help you understand how storage engines fit in to MySQL. (The NDB storage engine was added to MySQL just before this book was printed. Watch for it in the second edition.)

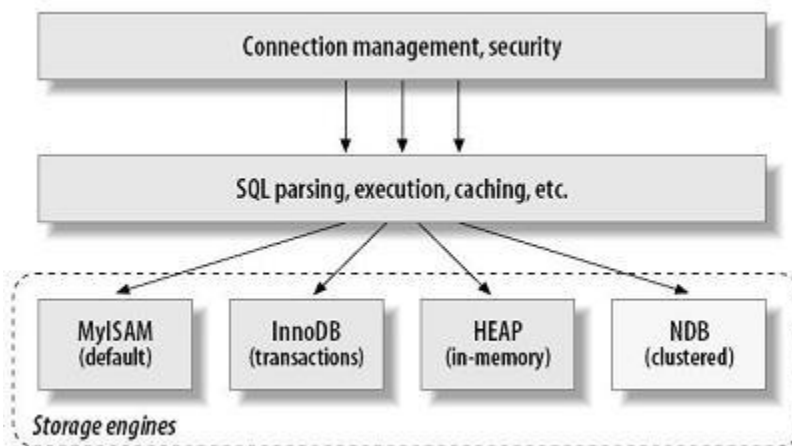


Figure 2-1. A logical view of MySQL architecture.

The topmost layer is composed of the services that aren't unique to MySQL. They're services most network-based client/server tools or servers need: connection handling, authentication, security, etc.

The second layer is where things get interesting. Much of the brains inside MySQL live here, including query parsing, analysis, optimization, caching, and all the built-in functions

The third layer is made up of storage engines. They're responsible for the storage and retrieval of all data stored "in" MySQL. Like the various file systems available for Linux, each storage engine has its own benefits and drawbacks. The good news is that many of the differences are transparent at the query layer.

The interface between the second and third layers is a single API not specific to any given storage engine. This API is made up of roughly 20 low-level functions that perform operations such as "begin a transaction" or "fetch the row that has this primary key" and so on. The storage engines don't deal with SQL or communicate with each other; they simply respond to requests from the higher levels within MySQL.

## **InnoDB**

InnoDB is a storage engine for MySQL, included as standard in all current binaries distributed by MySQL AB. Its main enhancement over other storage engines available for use with MySQL is ACID-compliant transaction support, similar to PostgreSQL, along with foreign key support (Declarative Referential Integrity).

InnoDB became a product of Oracle Corporation after their acquisition of Innobase Oy in October 2005. The software is dual licensed; it is distributed under the GNU General Public License, but can also be licensed to parties wishing to combine InnoDB in proprietary software.

## **Comparison with MyISAM**

MyISAM is the usual alternative engine, and the following are some of the properties that distinguish the two:

1. InnoDB recovers from a crash or other unexpected shutdown by replaying its logs. MyISAM must fully scan and repair or rebuild any indexes or possibly tables which had been updated but not fully flushed to disk. Since the InnoDB approach is approximately fixed time while the MyISAM time grows with the size of the data files perceived availability and

2. MyISAM relies on the operating system for caching reads and writes to the data rows while InnoDB does this within the engine itself, combining the row caches with the index caches. Dirty (changed) database pages are not immediately sent to the operating system to be written by InnoDB, which can make it substantially faster than MyISAM in some situations.

3. InnoDB stores data rows physically in primary key order while MyISAM typically stores them mostly in the order in which they are added. Storing the rows in primary key order corresponds to the MS SQL Server feature of “Clustered index” and the Oracle feature known as “index organized tables.” When the primary key is selected to match the needs of common queries this can give a substantial performance benefit. For example, customer bank records might be grouped by customer in InnoDB but by transaction date with MyISAM, so InnoDB would likely require fewer disk seeks and less RAM to retrieve and cache a customer account history. On the other hand, inserting data in orders that differ substantially from primary key (PK) order presumably requires that InnoDB do a lot of reordering of data to get it into PK order. This places InnoDB at a slight disadvantage, in that it does not permit insertion order based table structuring.

4. InnoDB currently does not provide the compression and terse row formats provided by MyISAM, so both the disk and cache RAM required may be larger. A lower overhead format is available for MySQL 5.0, reducing overhead by about 20% and use of page compression is planned for a future version.

5. When operating in fully ACID-compliant modes, InnoDB must do a flush to disk at least once per transaction, though it will combine flushes for inserts from multiple connections. For typical hard drives or arrays, this will impose a limit of about 200 update transactions per second. Higher transaction rates with maintaining transactional integrity is achievable with InnoDB leveraging ZFS file system with Flash disks, without the need for dedicated array hardware. Other methods include disk controllers with write caching and battery backup to maintain transactional integrity. InnoDB also offers several modes which reduce this effect, naturally leading to a loss of transactional integrity. MyISAM has none of this overhead, but only because it does not support transactions.

## **MyIsam**

MyISAM is the default storage engine for the MySQL relational database management system. It is based on the older ISAM code but has many useful extensions. In recent MySQL versions, the InnoDB engine has widely started to replace MyISAM due to its support for transactions, referential integrity constraints, and higher concurrency.

Each MyISAM table is stored on disk in three files. The files have names that begin with the table name and have an extension to indicate the file type. MySQL uses a .frm file to store the definition of the table, but this file is not a part of the MyISAM engine, but instead is a part of the server. The data file has a .MYD (MY Data) extension. The index file has a .MYI (MY Index) extension.

### **2.3.3 JAVASCRIPT**

Java Script is the scripting language of the Web. JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more. JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and JavaScript copies many Java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape bundling Sun's Java runtime with their then-dominant browser.[citation needed] The key design principles within JavaScript are inherited from the Self and Scheme programming languages.

"JavaScript" is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape Communications and current entities such as the Mozilla Foundation.

Due to the widespread success of JavaScript as a client-side scripting language for web pages, Microsoft developed a compatible dialect of the language, naming it JScript to avoid trademark issues. JScript added new date methods to fix the non-Y2K-friendly methods in JavaScript.

and "JScript" are often used interchangeably. Microsoft, however, notes dozens of ways in which JScript is not ECMA-compliant.

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- **JavaScript gives HTML designers a programming tool** - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- **JavaScript can put dynamic text into an HTML page** - A JavaScript statement like this: `document. Write("<h1>" + name + "</h1>")` can write a variable text into an HTML page
- **JavaScript can react to events** - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- **JavaScript can read and write HTML elements** - A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data** - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
- **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer.

## The Real Name is ECMA Script

- JavaScript's official name is ECMA Script.
- ECMA Script is developed and maintained by the ECMA organization.
- ECMA-262 is the official JavaScript standard.
- The language was invented by Brendan Erich at Netscape (with Navigator 2.0), and has appeared in all Netscape and Microsoft browsers since 1996.
- The development of ECMA-262 started in 1996, and the first edition of was adopted by the ECMA General Assembly in June 1997.
- The standard was approved as an international ISO (ISO/IEC 16262) standard in 1998.
- The development of the standard is still in progress.

## Syntax

The HTML `<script>` tag is used to insert a JavaScript into an HTML page.

```
<Html>  
<body>  
<scripttype="text/JavaScript">  
document.Write("HelloWorld!");  
</script>  
</body>  
</html>
```

To insert a JavaScript into an HTML page, we use the `<script>` tag. Inside the `<script>` tag we use the `type` attribute to define the scripting language. So, the `<script type="text/JavaScript">` and `</script>` tells where the JavaScript starts and ends.

## Browsers Support

- Browsers that do not support JavaScript will display JavaScript as page content.
- To prevent them from doing this, and as a part of the JavaScript standard, the HTML comment tag should be used to "hide" the JavaScript.
- Just add an HTML comment tag `<!--` Before the first JavaScript statement, and a `-->` (end of comment) after the last JavaScript statement, like this:
- JavaScript in the body section will be executed WHILE the page loads.
- JavaScript in the head section will be executed when CALLED.

JavaScript in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

### Scripts in `<head>`

Scripts to be executed when they are called, or when an event is triggered, go in the head section. If we place a script in the head section, you will ensure that the script is loaded before anyone uses it.

### Scripts in `<body>`

Scripts to be executed when the page loads go in the body section. If we place a script in the body section, it generates the content of a page.

## Using an External JavaScript

If we want to run the same JavaScript on several pages, without having to write the same script on every page, we can write a JavaScript in an external file. Save the external JavaScript file with a .js file extension.**Note:** The external script cannot contain the `<script>` tag! To use the external script, point to the .js file in the "src" attribute of the `<script>` tag

### 2.3.4 Hypertext Markup Language (Html):

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets.

HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML.

#### Basic HTML Tags

<code>&lt;! -- --&gt;</code>	specifies comments
<code>&lt;A&gt;.....&lt;/A&gt;</code>	Creates hypertext links
<code>&lt;B&gt;.....&lt;/B&gt;</code>	Formats text as bold
<code>&lt;BIG&gt;.....&lt;/BIG&gt;</code>	Formats text in large font.
<code>&lt;BODY&gt;...&lt;/BODY&gt;</code>	Contains all tags and text in the HTML document
<code>&lt;CENTER&gt;...&lt;/CENTER&gt;</code>	Creates text
<code>&lt;DD&gt;...&lt;/DD&gt;</code>	Definition of a term
<code>&lt;DL&gt;...&lt;/DL&gt;</code>	Creates definition list
<code>&lt;FONT&gt;...&lt;/FONT&gt;</code>	Formats text with a particular font
<code>&lt;FORM&gt;...&lt;/FORM&gt;</code>	Encloses a fill-out form
<code>&lt;FRAME&gt;...&lt;/FRAME&gt;</code>	Defines a particular frame in a set of frames
<code>&lt;H#&gt;...&lt;/H#&gt;</code>	Creates headings of different levels( 1 – 6 )

<code>&lt;HEAD&gt;...&lt;/HEAD&gt;</code>	Contains tags that specify information about a document
<code>&lt;HR&gt;...&lt;/HR&gt;</code>	Creates a horizontal rule
<code>&lt;HTML&gt;...&lt;/HTML&gt;</code>	Contains all other HTML tags
<code>&lt;META&gt;...&lt;/META&gt;</code>	Provides meta-information about a document
<code>&lt;SCRIPT&gt;...&lt;/SCRIPT&gt;</code>	Contains client-side or server-side script
<code>&lt;TABLE&gt;...&lt;/TABLE&gt;</code>	Creates a table
<code>&lt;TD&gt;...&lt;/TD&gt;</code>	Indicates table data in a table
<code>&lt;TR&gt;...&lt;/TR&gt;</code>	Designates a table row
<code>&lt;TH&gt;...&lt;/TH&gt;</code>	Creates a heading in a table.

### **3. SYSTEM STUDY AND ANALYSIS**

System study and analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes.

#### **3.1 Existing System**

The existing system is not flexible as all the procedures with respect to both the candidate and the recruiter is done manually. This increases the paper work of the company which includes the application by the candidate for the job, the resumes submitted etc. The work of the consultancy is better noticed online rather than banners or any other mode.

The candidate goes to the consultancy directly and applies for a job is time-consuming. The other constraint is keeping the track of all the resumes at a time. Thus, there arises confusion because it will be intricate to check all the resumes and notify the requirement for a certain job. Therefore the existing system is

- Not flexible
- Involves large amount of paper work
- Time consuming process
- Slow Updating and Retrieval of Information.
- Cost huge And Unpredictable
- Manually screening the applicant resumes will be time consuming. Maintaining each applicant's data with much number of fields is a tedious process and takes more time to retrieve back.

## **3.2 Proposed System**

The proposed system eases the work of the HR recruiter and also provides utmost support to the candidate applying for the job. The candidates can get to know about the requirements. This also helps to know the candidates who were short-listed by the company. This helps the company to place qualified and eligible candidates. The requirements can put up online by the company and the resumes could be sent to the company simultaneously. Therefore the proposed system is:

- Involves faster processing of data.
- Flexible
- Involves less paperwork.
- Faster information retrieval and updating.
- Report Generation.
- Reduces Cost Effectively.
- Competition for the most talented individuals is increasing. Organizations need to secure the best people quickly and efficiently to help them build competitive capability.

## **3.3 Feasibility Study**

The feasibility study is conducted to verify the benefits of the proposed system. The feasibility study also analyses the requirements of the proposed system. Three types of feasibility study are carried out. They are Economical Feasibility, Technical Feasibility and Operational Feasibility. The cost, technology in hardware and software, operational and social impacts is considered in the feasibility study.

### **3.3.1 Economic Feasibility**

The economic feasibility is conducted to analyze the cost benefits of the system. The development cost, installation cost, infrastructure cost and operational cost factors are analysed in the economic feasibility study. The development cost includes the application development cost. The operational cost is analyzed to maintain regular expenses and the cost benefit difference between the existing system and proposed system.

- Tangible benefits: - Saving man labor to do tedious tasks saves time.
- Intangible benefits:-Improves the quality of organization.

### **3.3.2 Operational Feasibility**

The operational feasibility study is conducted to analyze the operational changes between the existing system and the proposed system. The operational change and training details are also analyzed in the operational feasibility analysis. The system support easy execution procedures. The system can also provide easy menu driven models. Anyone with minimum computer knowledge can access the system. The system does not require any special operational training.

### **3.3.3 Technical Feasibility**

The technical feasibility study is carried out to assess the hardware and software technology is required for the system. The hardware requirement and its availability are analyzed in the hardware verification process. The system does not require any additional hardware peripherals to execute the system. The PHP is used as the development environment for the system. The system uses the SQL Server 2008 as back end.

## **4. SYSTEM DESIGN**

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical system design. The inputs to the design are the software requirements and the output will be the design specification applicable to software.

### **4.1 Input Design**

Input design is a very essential part in the project and should be concentrated well as it is prone to error. The data that are to be inserted plays an important role. In order to get the meaningful output and to achieve good accuracy, the input should be acceptable and understandable. It is the process of converting user-originated inputs to a computer-based format and it is one of the most expensive phases of the operation of computerized system and is often major problem of a system. In this project the input designs are made in various web forms with various methods to carry out the process which is used in the project. System analyst decide the following input design details like, what data item to input, what medium to use, how the data should be arranged or coded data items and transactions needing validations to detect errors and at last the dialogue to guide users in providing input.

- The data to be entered by the user in every web form.
- The way of data arrangement in web forms.
- The messages to guide the users in providing input.
- The Data items and transaction needs validation to detect errors.
- Methods for performing input validation and steps to follow when error occurs.

In this project, the input design is made in various web forms using HTML, PHP coding and CSS tags.

Contrivance planning which hold the common user interface functionality for the entire application. It uses the html controls like list and also uses the cascading style sheet properties margin, font, border, float, list-style-type, padding, height, width, background-color, text-decoration to enhance the user interface functionality.

The web forms also uses the web user interface controls like label, text box, drop down box, button. Validation control like required field validation and validation group is used to validate the user inputs.

## **4.2 Output Design**

Output Design generally refers to the results and information that are generated by the system for many end-users output is the main reason for developing the system and on the basis of which they evaluate the usefulness of the application.

The success and failure of the system depends on the output. The outputs generated by the system are checked for its consistency. In any system the output design determines the input to given to the application. The objective of a system finds its shape in terms of the output. The most common are screen displays, and printed form, graphical drawing etc. The outputs also vary in terms of their contents, frequency, timing and format. The basic requirements of outputs should be accurate, timely and appropriate in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner.

Output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

**When designing the output, system analyst must accomplish the following**

- Determine the information to present in every web form.
- Decide whether to display information in grid view or tables.
- Arrange the information in user friendly manner.
- Decide how to distribute the output to intended receipt.

In this project, the output mainly refers to the web forms generated. The data from the database is displayed in the grid view.

The output of this project displays,

- Requirement specification of the companies
- Details of the candidate requirements.
- The scheduled interview details and the venue.
- The status of the selected candidate.
- Leave summary and payment details of the employees.
- Records of the employee details.
- Billing details of both the client and candidate .

## **5. SYSTEM DEVELOPEMENT**

### **5.1 Modules**

This project mainly has four modules and are

1. Administration Module.
2. Employee Module.
3. Client Module.
4. Candidate Module.

### **5.2 Module Description**

#### **5.2.1 Administration Module**

In Administration Module, the admin has all the rights to use the application. Admin will register the employee details and will provide the username and password for the employee, so that they can access the site. Every employee has different employee ID. The admin will maintain the payroll and leave record of the employee.

Admin can delete the records of the employee who left out the organization. Admin can see all the records and details of the employee, Client Company and candidate. The admin have the right to delete the records. The different sub modules in this module are:

##### **➤ Employee Registration**

This sub module can be used to add new employee to this organization. The admin will register the new employee. Basic details given by the employee about himself will be verified by the admin. Once the employee is registered, the admin will provide a user name and password for the employee. Once the employee is activated, the admin will provide the existing features to the employee. Only by the username and password provided to the employee, the employee can access the site and can do the work. The information of the employee will be maintained confidential by the administration.

➤ **Update/Delete Employee Details**

This sub module helps to make any modifications to existing employee record and update it or deletes the specified employee record from the database. If any employee's address or phone number changes or their designation got changed, the admin can update their records in this sub module. If the employee leaves the organization the admin can delete his record from the database. And the username and password of that employee will be blocked so that he cannot access the site again.

➤ **Delete Client Details and Requirements**

In this sub module the admin can see the client details and client requirements and even the admin can delete the details and requirement of the client. The admin cannot update the client details and requirements.

➤ **Delete Candidate Details and Resume**

In this sub module the admin can see the candidate details and resume. The admin can delete the details and resume of the candidate. The admin cannot update the candidate details and resume.

➤ **Payroll and Leave Management**

In this sub module the admin will maintain the employee's payroll and leave details. Only the admin has the right to see this and make the changes in the particular employee's salary.

## **5.2.2. Employee Module**

Once the employee gets the username and password, he got the rights to access the employee module. By login with the username and password he will get in to the employee module and he can use the features provided in the employee modules. The employee can see the requirements of the clients. And the employee also has right to see the candidate resume.

➤ **Selection module**

The selection module is concerned with the understanding of the job requirement and the candidate's qualification and skills.

The candidate qualification is scrutinized and the suitable person is shortlisted. The selection process is vital to the organization as it deals with the proper analysis of the qualification as well as his skill sets which mean a lot to the recruiting company. This helps to reduce the burden of the clients in searching candidates and will help to post qualified candidates within the time limit.

➤ **Interview module**

In this module the short listed candidates, who are qualified for the particular position, are sent to the respective company for the different terms of interview and group discussions. The time, venue, date and the attested forms are some important documents which need to be taken by the candidates to the specified company. All the details of the particular candidate and the company will be given in this module. And it will be also mentioned that which candidate is selected for which round in the particular company.

➤ **Billing module**

In Billing Module Soon after the candidates skill sets is scrutinized by the company themselves they can come to a conclusion regarding the candidates to be selected. As soon as the consultancy is intimated about the employment for the particular candidate the formalities regarding the commission is done. This constitutes the billing module. The necessary amount to be given to the consultancy, the billing and the other procedures are carried out.

### **5.2.3 Client Module**

In this module the client needs to register on the site. Only then he will be able to access the site. Once the client got registered on the site his account gets activated, then he can access the client module. In this module the client can put his requirements and the contact details.

➤ **Requirement Module**

In this sub module the client company can put their requirement on the site. On the basis of those requirements the consultancy will select the candidate for the interview. And thus the candidates have the choice of applying for the particular positions he would like to acquire. The process in all provides job satisfaction to the candidates, as well as assurance to the company in posting the eligible candidates.

➤ **Client Contact Module**

In this sub module the client company can put the details of the the contact person and the department of the contact person where the consultancy person and the selected candidates need to contact.

➤ **Interview module**

In this sub module, the client company will put the information about the candidates who are qualified for the next round if the interview in the particular company. And it will also be mentioned that which candidate is selected for which round in the particular company.

## **5.2.4 Candidate Module**

In this module the candidate needs to register on the site. Only then he will be able to access the site. Once the candidate got registered on the site and his account gets activated, then he can access the candidate module. The candidate can put his resume and he can update and delete the resume. In this module the candidate is also able to see that whether he is selected for the interview or not.

➤ **Submit Resume**

In this module, the candidate needs to submit his resume. He need to fill all the required fields and he has to attach his resume with it. On the basis of that the selection criteria will be carried out. On the basis of the candidate resume the consultancy will select the candidate for the interviews.

➤ **Update/Delete**

In this sub module, the candidate can update his resume or if he wants to delete the resume then he can delete the resume also. In this module the candidate is also able to update his contact details.

➤ **Search**

In this sub module, the candidate is able to search that whether he is selected for any round or not. In this the candidate can see only his information and he is not able to see the information of the other candidate.

## 6. SYSTEM TESTING AND IMPLEMENTATION

### 6.1 System Testing

**Testing** is a process used to help identify the correctness, completeness and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software.

The quality issues of the application vary widely from system to system but some of the common quality attributes include reliability, stability, portability, maintainability and usability. Refer to the ISO standard ISO 9126 for a more complete list of attributes and criteria. Testing helps in verifying and validating if the Software is working as it is intended to be working. This involves using Static and Dynamic methodologies to test the application.

#### **Testing Objectives Include**

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as yet undiscovered error.
3. A successful test is one that uncovers an as yet undiscovered error.

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications.

The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

Testing early in the life cycle reduces the errors. Test deliverables are associated with every phase of development. The goal of **Software Tester** is to find bugs, find them as early as possible, and make them sure they are fixed.

The number one cause of Software bugs is the **Specification**. There are several reasons specifications are the largest bug producer.

In many instances a Spec simply isn't written. Other reasons may be that the spec isn't thorough enough, it's constantly changing, or it's not communicated well to the entire team. Planning software is vitally important. If it's not done correctly bugs will be created.

The next largest source of bugs is the **Design**, That's where the programmers lay the plan for their Software. Compare it to an architect creating the blue print for the building, Bugs occur here for the same reason they occur in the specification. It's rushed, changed, or not well communicated.

**Coding errors** may be more familiar to you if you are a programmer. Typically these can be traced to the Software complexity, poor documentation, schedule pressure or just plain dump mistakes. It's important to note that many bugs that appear on the surface to be programming errors can really be traced to specification. It's quite common to hear a programmer say, "Oh, so that's what it's supposed to do. If someone had told me that I wouldn't have written the code that way."

The other category is the catch-all for what is left. Some bugs can be blamed for false positives, conditions that were thought to be bugs but really weren't. There may be duplicate bugs, multiple ones that resulted from the square root cause. Many modern software applications are so complex, and run in such as interdependent environment, that complete testing can never be done. "When to stop testing" is one of the most difficult questions to a test engineer. Common factors in deciding when to stop are:

- Deadlines (release deadlines, testing deadlines.)
- Test cases completed with certain percentages passed
- Test budget depleted
- Coverage of code/functionality/requirements reaches a specified point
- The rate at which Bugs can be found is too small

- Beta or Alpha Testing period ends
- The risk in the project is under acceptable limit.

Practically, we feel that the decision of stopping testing is based on the level of the risk acceptable to the management. As testing is a never ending process we can never assume that 100 % testing has been done, we can only minimize the risk of shipping the product to client with X testing done. The risk can be measured by Risk analysis but for small duration / low budget / low resources project, risk can be deduced by simply:

- Measuring Test Coverage.
- Number of test cycles.
- Number of high priority bugs

## **Testing Methodologies**

The following are the Testing Methodologies:

1. Unit Testing.
2. Integration Testing.
3. User Acceptance Testing.
4. Output Testing.
5. Validation Testing.

## **Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

## **Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction.

After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design. The following are the types of Integration Testing:

➤ **Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breath first manner.

➤ **Bottom-Up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

## **User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly in touch with the prospective system users at time of developing and making changes wherever required is done in regard to the following point:

- Input Screen design.
- Output Screen design

## **Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. The outputs generated or displayed by the system under consideration are tested by asking the users about the format required by them. Hence the output format is considered in 2 ways – one is on screen and another in printed format

## **Validation Checking**

Validation checks are performed on the following fields.

### **➤ Text Field**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

### **➤ Numeric Field**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

## **Test Plan**

- 1) **Title of the Project: “Contrivance Planning Management System”**
- 2) **Objective of the document:** In this test plan we are covering the activities and functionality of different modules and their sub modules. In this document we covering what kind of test cases should described.

3) **Scope of the document:** In this document, in each phase, what are going to do and how are going to do. In this section we are mentioning which requirements are testing. Take one module i.e. user login module. For user module the client requirements are:

- User name must be not less than 4 character and not greater than 10.
- Username must start with character, not with digits.
- Username should not contain any special character.
- Password field should fill with at least 4character.

The above requirements are only for login module and other functional requirements can declared in this section. The requirements which are declared in this section it may depend on other requirements on different module.

4) **Objective of Testing:** The main objective of testing in this application is to chances of preventing the defects on the client environment.

5) **Critical Functionalities:** In this section we are discussing Key roles & Causes for success of application.

6) **Test Data Requirements and Collection:**

In this section we are collecting the requirements for the application. From the different resources. Such as

- Collecting from the client.
- Refereeing the existing applications which are similar to current one's

7) **Training Requirements:**

Training requirements are focused in 2 areas

- a) Technology – in this we are discussing about the new technology in the market and scope of that technologies in the future.
- b) Domain- in this we are discussing about the Existing personal Training knowledge.

8) **Resource Requirements:** In this we discussing about the required resources for the application such

- Employees
- Billing Number

9) **Scheduling:** In this section we will be discussing about Staring Date and End Date of the application are given.

10) **Input Criteria:** There are different criteria conditions i.e. Unit Testing, Release Note & Installation.

11) **Exit Criteria:** In this only evaluation documents only accepted.

12) **Risk Analysis:** In this we are converse about Risk Analysis such as

- a. Risk on Resources.
- b. Risk on TimeLine.
- c. Risks are identified by Preparing Solution Plan.

## Types Of Tests Carried

In our application we are performing the Gray Box testing (white and black).

Test No.	Form Name	Inputs to different fields	Expected Result	Actual Result	Status	Remarks
1	Employee Registration	Employee ID="EE0"	Error Message: Enter valid Employee ID	Enter valid Employee ID	Test Success	Nil
2	Candidate Registration	Candidate Name="Ram"	No Error	No error message displayed	Test Success	Nil
3	Client Registration	E-Mail = abc@gmail.com	Error Message: Enter valid Email-id	No error message displayed	Test failed	As you have to enter valid email-id
4.	Payroll	Salary="12000"	No error	No error message displayed	Test Success	Nil
5	Placement	Date of Joining "201-12-25"	Error Message: Date Format is not correct	No error message displayed	Test Failed	You have to enter in the format (yyyy-mm-dd)
6	Requirement	No. of requirements="5"	No error	No error message displayed	Test Success	Nil
7	Candidate Registration	Contact Number="973344 223"	Error: Enter the valid contact number	Error message displayed	Test Success	Nil
8	Employee Registration	Address="&12, ads, bangalore"	Error message: Enter the valid address	Error message displayed	Test Success	Nil

## **6.2 System Implementation**

Implementation is a process of bringing a developed system into operation and turning it over to the user. Implementation activities extend from the number of plans. The beginning of the phase is to create the schedule and manage the different activities that must be integrated into plan.

Implementation is the stage of the project when the theoretical design is turned in to a working system. The most crucial stage is achieving a successful new system and is giving the user's confidence that the new system will work and be effective in the implementation stage.

In the implementation phase, first, forms are created to begin the process, as soon as the forms are created the values are given as an input to check whether the modules are executing properly after checking the process the results will be displayed. Thus the implementation phase is completed successfully.

## **6.3 System Maintenance**

The term “software maintenance” is used to describe the software engineering activities that occur following delivery of a software product to the customer. The maintenance phase of the software life cycle is the time period in which a software product performs useful work. Maintenance activities involve making enhancement to software products, adapting products to new environments and correcting problems. Software product enhancement may involve providing new functional capabilities, improving user display and modes of interaction, and upgrading external documents. Adaptation of software to a new environment may involve moving the software to a different machine. Problem correction involves modification and revalidation of software to correct errors. The enhancement of this project can be accomplished easily. Adaptation of this project to a new environment is also performed easily.

### **6.3.1 Corrective Maintenance**

Even with the best quality assurance activities, it is likely that their customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

### **6.3.2 Adaptive Maintenance**

An activity that modifies the software to the properly interface with a changing environment. The system has been modified so that various changes include to the new system.

## 7. CONCLUSION

The project “**Contrivance Planning Management System**” is successfully completed in time with all deliverables meeting the requirements which provide the information that is needed to manage organizations efficiently and effectively. The system objectives were fulfilled with all the requirements specified by the analysis which was completed satisfactorily.

This software will be launched for the internal use of company and it is efficiently handles the complete process and is satisfied with performance of the system. This system is fully functional, easy to use and no user training or orientation necessary. The screens are GUI based data entry screens and very interactive, proper status messages are displayed and hence easy for user to interact and understand.

By using these features of PHP this project has proved to be successful in performing complex diagnostics, route data between screens and generate required responses. There are many tangible and intangible benefits in implementing the system. Thus the project was successfully developed and completed using PHP and MYSQL server as the back end.

This system encompass three primary components: Technology, People (individuals, groups, or organizations), and Data/Information for decision making. Our systems are distinct from other information systems in that they are designed to analyze and facilitate strategic and operational activities in the organization. All the Modules of this application are successfully developed and implemented.

## **8. SCOPE FOR FUTURE ENHANCEMENT**

As the times keeps on going, the requirements also increase. Every client and candidate needs more facilities. Hence the requirements of the clients and candidates also increase. To meet the requirement of the clients and the candidates we can do the enhancement in this project.

The future enhancements that can be done are as follows:

- Online Fresh Resume preparation.
- Online test for candidates as per the client requirements.
- Online courses as per the industries requirement.

## 9. BIBLIOGRAPHY

### For PHP

- PHP5 and MySQL by W Jason Gilmore (Apress)
- PHP Web Development by Allan Kent and David Powers (Apress)
- Professional PHP4 by Christopher Scollo
- PHP 5 Unleashed by John M. Coggeshall
- Advanced PHP Programming by George Schlossnagle
- Programming PHP by Kevin Tatroe
- Zend PHP Certification Study Guide by Zend Technologies

### For MySQL

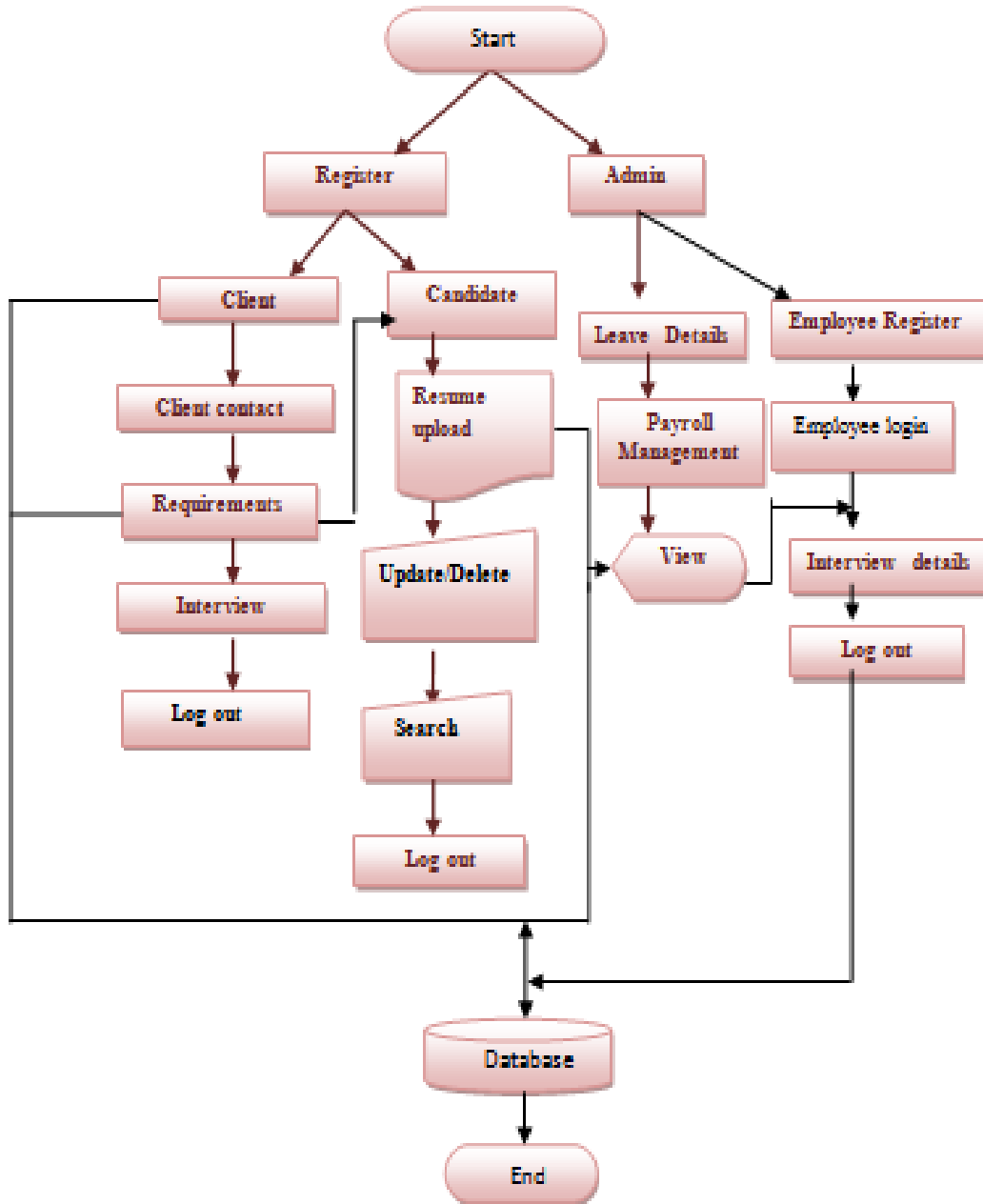
- MySQL (2nd Edition) (Developer's Library) by Paul Dubois
- Php and MySql---Larry Ullman

### For JAVASCRIPT

- <http://www.w3schools.com/>
- <http://www.zend.com/php/beginners/>
- <http://www.devshed.com/c/b/PHP/>
- <http://www.phpfreaks.com/>

## 10. APPENDIX

### A) Flow Chart

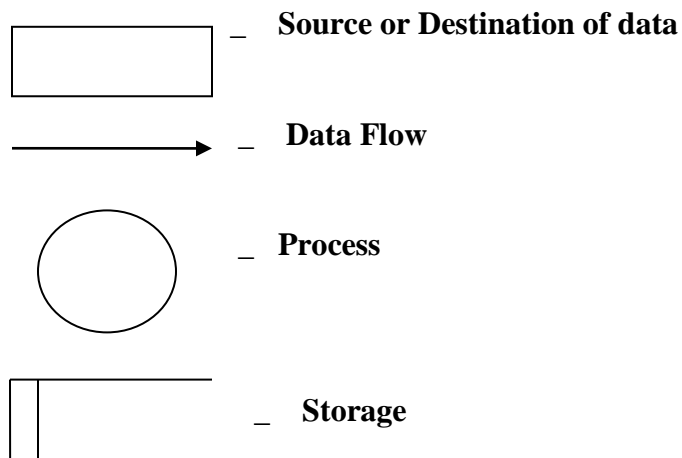


## B) DATA FLOW DIAGRAM

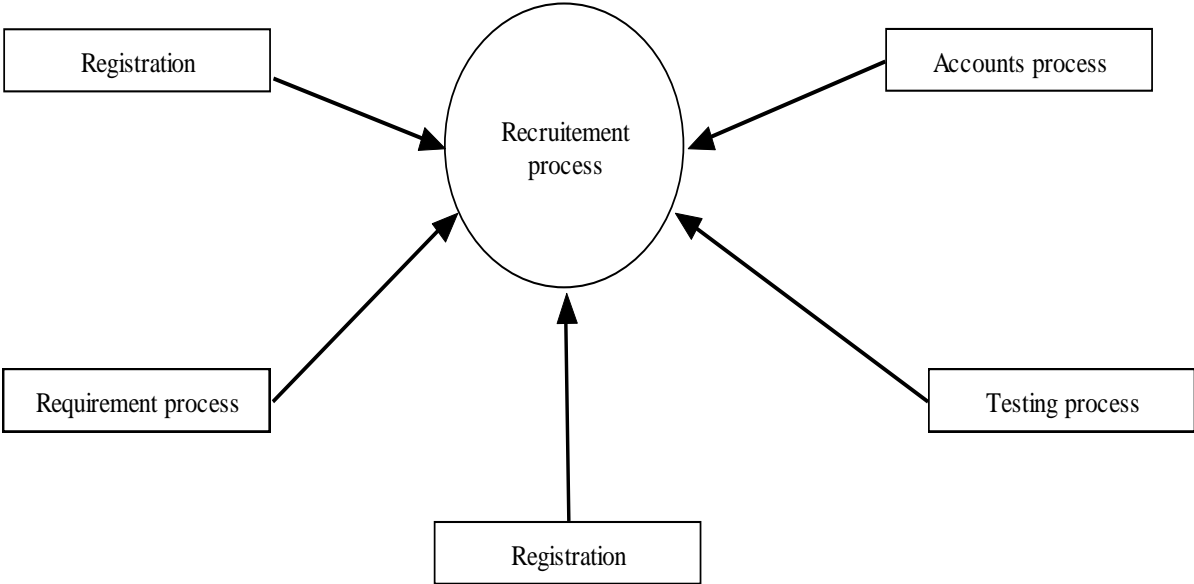
A **data-flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

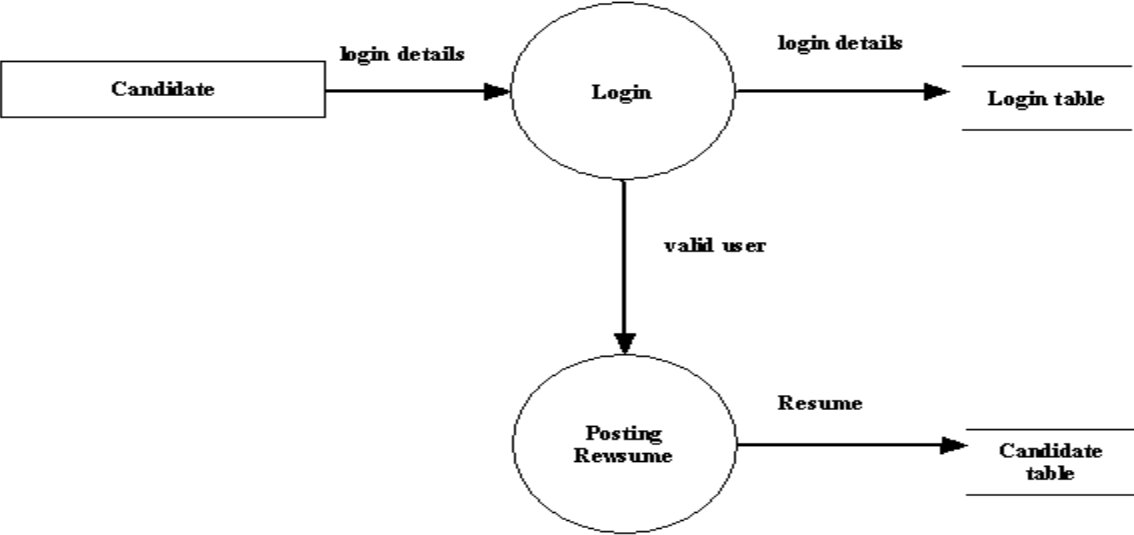
### Data Flow Diagram Symbols



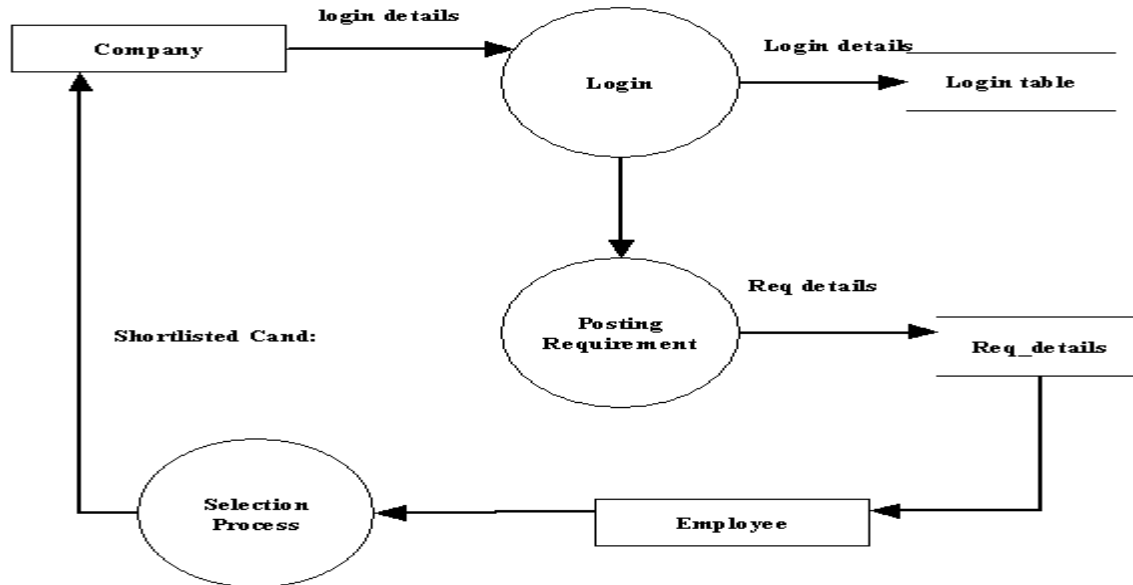
# Context Analysis Diagram



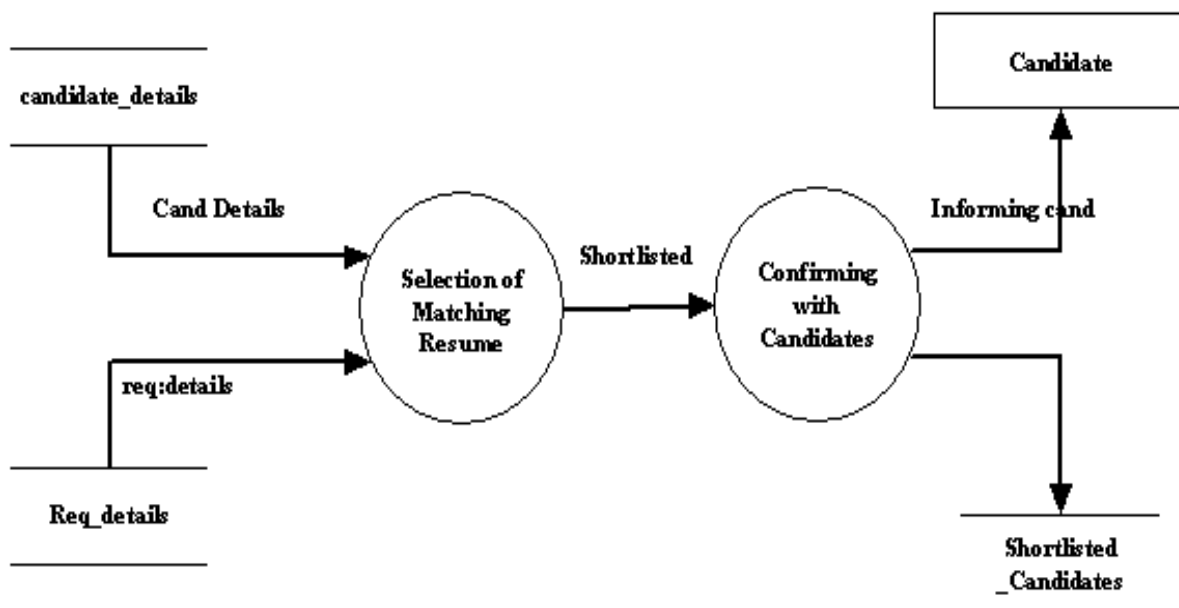
# Level 0: Candidate



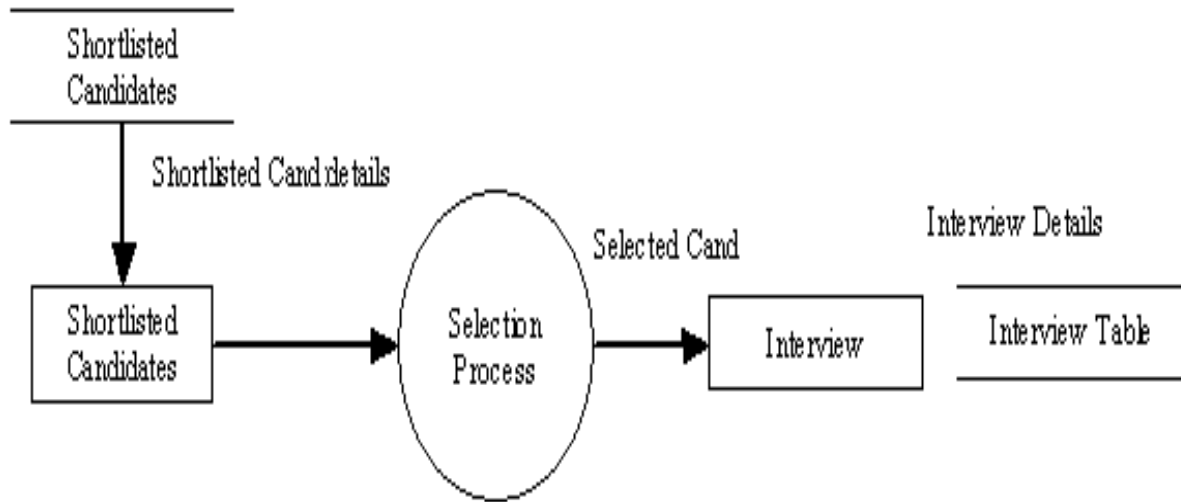
## Level 1: Client



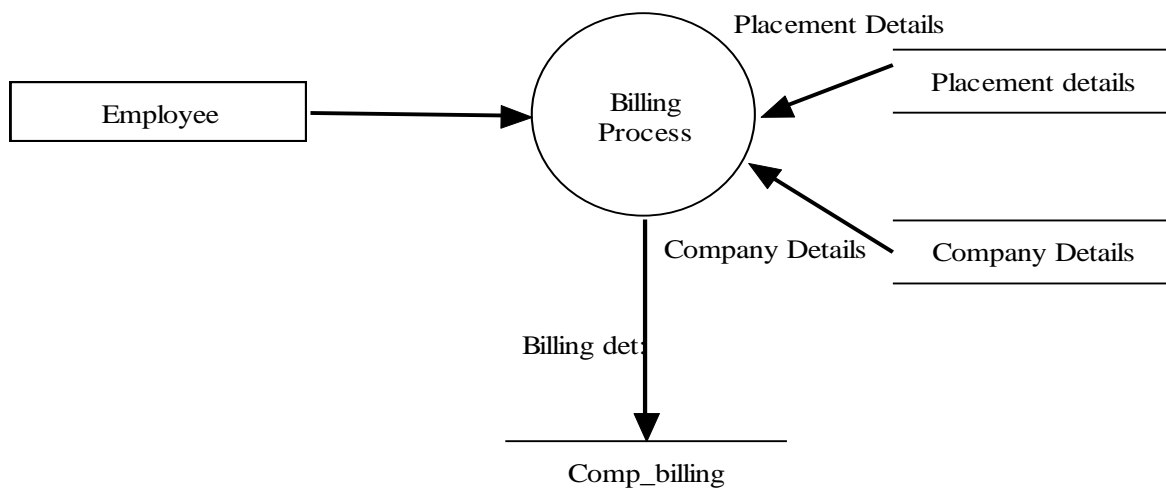
## Level 2: Selection Process1



### Level 3: Selection Process2



### Level 4: Billing



## c) SCREEN SHOTS

### Screen C1: Home Page



### Screen C2: Client Login



## Screen C3: Client Requirements

Web Job Portal For IT Concern

### Contrivance Planning Management System

Client Requirement

Comp_ID	Skills	Location	Job Function	Required Date	Position	No of Requirement	Experience
Soft21	JAVA	CBE	development	2012-03-10	Developer	2	2
softw12	PHP	coimbatore	develper	2014-02-12	gandhipuram	5	4

BACK

## Screen C4 :Candidate Details

Web Job Portal For IT Concern

### Contrivance Planning Management System

Candidate Details

ID	Candidate ID	First Name	Last Name	Gender	Address	City	Username	Password	Email ID	Contact Number
1	CAND1	siva	kumar	male	CBE	CBE	siva	Siva123	siva@gmail.com	1234567890
2	CAND2	vaishu	manimozhi	male	pollachi	coimbatore	priya	kumar	vaishnavee68@gmail.com	8989087779
3	CAND3	priyadharshini	krishnan	female	Ramalingam colony	Coimbatore	krishnan	krishna	krishna7128@yahoo.com	8974324105

DELETE    BACK

## Screen C5: Candidate Page

Web Job Portal For IT Concern

### Contrivance Planning Management System

**Candidate Resume**

Candidate ID	<input type="text" value="CAND2"/>	Email Id	<input type="text" value="navee689@gmail.com"/>
Qualification	<input type="text" value="MCA"/>	Mobile Number	<input type="text" value="8973632105"/>
Specialisation	<input type="text" value="JAVA"/>	Phone Number	<input type="text" value="04259244267"/>
Language Known	<input type="text" value="ENGLISH"/>	Contact Address	<input type="text" value="1/49 pollachi"/>
Location	<input type="text" value="CBE"/>	Permanet Address	<input type="text" value="1/49POLLACHI"/>
Experience	<input type="text" value="2"/>	Attach Resume	<input type="button" value="Browse..."/> RESUME.docx

Copyright © Contrivance Planning Management System

## Screen C6: Admin Page

Web Job Portal For IT Concern

### Contrivance Planning Management System

<a href="#">Client</a>	<a href="#">Candidate</a>	<a href="#">Employee</a>	<a href="#">New Employee</a>	<a href="#">Logout</a>
------------------------	---------------------------	--------------------------	------------------------------	------------------------



## Screen C7 : Leave management

Web Job Portal For IT Concern

### Contrivance Planning Management System

**Leave Management**

Emp ID: EMP1

Total Leave:

Leave Taken:

Remaining Leave:

### Leave List

Emp Id	Total Leave	Leave Taken	Remaining Leave
EMP1	5	3	2
EMP1	4	2	

## Screen C8: Password Retrieval

Web Job Portal For IT Concern

### Contrivance Planning Management System

**Password Retrieval**

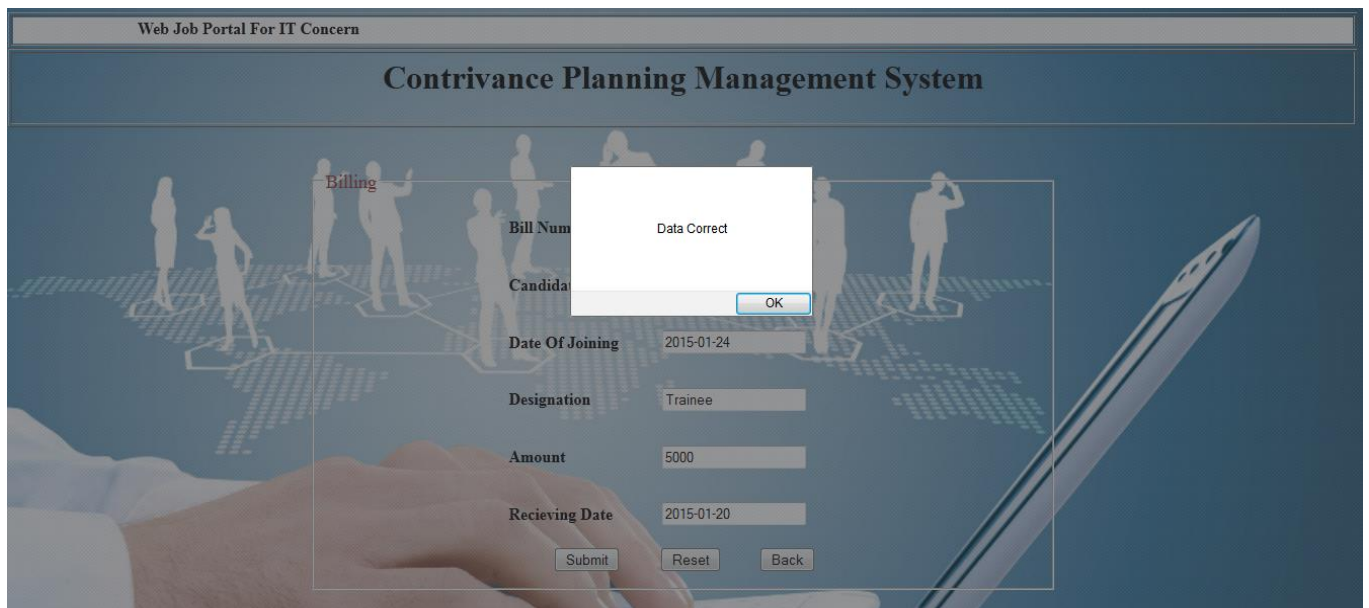
Employee Name:

Email Id:

## Screen C9: Employee Page



## Screen C10: Candidate Billing



## D) Table Design

### Table D1: Login

TEXT FIELD	DATA TYPE	DESCRIPTION
Username	Varchar(20)	Username
Password	Varchar(20)	Password

### Table D2: Employee Registration

TEXT FIELD	DATA TYPE	DISCRIPTION
emp_id	Varchar(10)	Employee ID(PK)
emp_name	Text	Employee Name
Age	Varchar(10)	Employee Age
Gender	Char(6)	Gender
emp_address	Varchar(30)	Employee Address
emp_city	Text	Employee City
emp_mobile	Decimal(10)	Employee Mobile No.
emp_mail	Varchar(30)	Employee Email ID
Username	Varchar(20)	Username
Password	Varchar(20)	Password
conf_password	Varchar(20)	Confirm Password
emp_degree	Varchar(20)	Employee Degree
emp_doj	Date	Date Of Joining
emp_yoe	Varchar(10)	Year Of Experience
emp_expertise	Text	Expertise in
emp_officemail	Varchar(30)	Official Mail ID

**Table D3: Client Registration**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DISCRIPTION</b>
comp_id	Varchar(10)	Company ID(PK)
comp_name	Varchar(30)	Company Name
address1	Varchar(30)	Address
address2	Varchar(30)	Address
city	Text	City
state	Text	State
pincode	Decimal(6)	Pincode
Username	Varchar(20)	Username
Password	Varchar(20)	Password
conf_password	Varchar(20)	Confirm Password
Email	Varchar(30)	Company Email ID
phone	Decimal(12)	Company's Phone No.
fax	Decimal(12)	Company's Fax No.

**Table D4: Candidate Registration**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DISCRIPTION</b>
cand_id	Varchar(20)	Candidate ID(PK)
Fname	Text	Candidate First Name
Lname	Text	Candidate Last Name
Gender	Char(6)	Gender
Address	Varchar(30)	Candidate Address
City	Text	City
Username	Varchar(20)	Username
password	Varchar(20)	Password
conf_password	Varchar(20)	Confirm Password
cand_email	Varchar(30)	Candidate Email ID
Contactno	Decimal(12)	Candidate Contact No.

**Table D5: Candidate Resume**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DISCRIPTION</b>
cand_id	Varchar(20)	Candidate ID(FK)
Qualification	Varchar(20)	Candidate Qualification
Specilisation	Varchar(20)	Specialised In
Language	Varchar(30)	Language known
location	Varchar(30)	Candidate Location
Experience	Varchar(10)	Candidate Experience
Email	Varchar(30)	Candidate Email ID
Mobile	Decimal(10)	Candidate Mobile No.
Phone	Decimal(12)	Candidate Phone No.
Address	Varchar(30)	Current Address
Permanent	Varchar(30)	Permanent Address
Resume	Varchar(100)	Candidate Resume

**Table D6: Candidate Billing**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DISCRIPTION</b>
Billno	Decimal(10)	Bill Number(PK)
cand_id	Varchar(20)	Candidate ID(FK)
Doj	Date	Date of Joining
Designation	Varchar(20)	Candidate Designation
Amount	Decimal(10)	Amount paid
Received	Date	Received Date

**Table D7 : Client Contact**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
comp_id	Varchar(10)	Company ID(FK)
Person	Text	Contact Person
Department	Varchar(30)	Department
Email	Varchar(30)	Company Email ID
Fax	Decimal(12)	Company Fax Number

**Table D8: Client Requirement**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
comp_id	Varchar(10)	Company ID(FK)
Skill	Varchar(30)	Skill Required
Location	Text	Company Location
Jobfunction	Text	Job Function Area
Reqdate	Date	Required Date
Position	Varchar(30)	Designation
n_o_r	Decimal(5)	No. of Requirements
Experience	Varchar(6)	Experience Required

**Table D9: Client Billing**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
Billno	decimal(10,0)	Bill Number(PK)
comp_id	varchar(10)	Company ID(FK)
Billdate	Date	Bill Date
Amount	decimal(10,0)	Total Amount
Duedate	Date	Due Date
Received	decimal(10,0)	Received Amount

**Table D10: Employee Payroll**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
emp_id	Varchar(10)	Employee ID(FK)
Salary	Varchar(10)	Employee Salary
Bonous	Varchar(10)	Bonous
Post	Varchar(20)	Designation
Allowance	Varchar(10)	Other Allowances
Net	Varchar(10)	Net Salary

**Table D11: Employee Leave**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
emp_id	Varchar(10)	Employee ID(FK)
tot_leave	decimal(2,0)	Total Leave
Ltaken	decimal(2,0)	Leave Taken
Remaining	decimal(2,0)	Remaining Leave

**Table D12: Interview**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
cand_id	Varchar(10)	Candidate ID
comp_id	Varchar(10)	Company ID
Interview	Date	Interview Date
Location	Text	Company location
Person	Varchar(30)	Contact person
Short	Varchar(12)	Shorted listed for
Result	Varchar(15)	Result

**Table D13: Placement**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
cand_name	Varchar(30)	Candidate Name
Designation	Varchar(30)	Designation
comp_name	Varchar(30)	Company Name
Doj	Date	Date of Joining
Salary	Decimal(10)	Salary

**Table D14: Contact Us**

<b>TEXT FIELD</b>	<b>DATA TYPE</b>	<b>DESCRIPTION</b>
Name	varchar(30)	Person Name
Email	varchar(30)	Email ID
Subject	varchar(30)	Subject
Comment	varchar(200)	Comments