Galgotias University

School of Computing Science & Engineering

Plot No. 2 Scetor 17- A Yamuna Expressway Greater Noida, Gautam Budh Nagar, Uttar Pradesh, 201310

"DIGITAL VOTING"

Submitted in partial fulfillment of the requirement for
The award of degree of

MASTER OF COMPUTER APPLICATION
(2019-2020)

SUBMITTED BY:-
ASHOK KUMAR (18032030084)

RANGEELA KUMARI (18032030082)

Semister :- 6th

GUIDED BY:-

Dr. S. Prakash (Assoc. Prof.)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"DIGITAL VOTING"** is the bonafide work of **"**
ASHOK KUMAR(18032030084) & RANGEELA KUMARI(18032030082)"
who carried out the project work under my supervision.

SIGNATURE OF HEAD   **SIGNATURE OF SUPERVISOR**

Dr. Munish Shabarwal   Dr. S.Prakash

PhD (Management), PhD(CS)   PhD, (Assoc. Prof.), SCSE

Prof. & Dean, SCSE

ACKNOWLEDGMENT

I wish to record my deep sense of gratitude and profound thanks to my research supervisor **Mr**. **S. Prakash,** Associate Professor, School of Computing Science and Engineering department, Galgotias University, Greater Noida for his keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this dissertation into fruition.

I extend my sincere thanks to Dean SCSE for providing excellent platform and resources to carry out my research projects. Also, I would like to thank the panel members for their valuable suggestions and support during presentation of my research projects.

Finally, I extend my sincere thanks to the University Management, All faculty members, non-teaching staff members and Lab Assistants of the SCSE Department, Galgotias University, Greater Noida, for their valuable support throughout the course of my **MCA** Dissertation.

I thank my friends, fellow researchers and family members who have encouraged me in my research efforts and shouldered me in needy times.

THANK YOU

Ashok Kumar

Rangeela Kumar

6TH SEMESTER

DECLARATION

We hereby declare that, this **Project** entitled **"Digital Voting"** and the work described in it is my own work, and are carried out in accordance with the regulations of the **Galgotias University**. The work is original except, where indicated by special references in the text and no part of the project has been submitted for any other degree.

Ashok Kumar
Rangeela Kumari
6TH SEMESTER

INDEX

1.INTRODUCTION ABOUT THE PROJECT

Digital Voting is a portal that is proposed to automate the functionality of voting. Currently we go polling center for voting but what about physically challenged voters and they faced very difficulty to complete this process so this project also focus this issue.

There are multiple advantages for this portal:

1. Transparency.
2. Time saving
3. Voting from other countries.
4. No need of Police /Army in the election duty.
5. Normal Transportation facility during voting
6. Fraud Detection
7. **Registration Process:**

   Through this module new posters can registered them. After giving their details, they will get a user id and password. Then to get entry into poster section they need to provide this id and password and only poster with valid id and password will get entry into poster zone. This is also a security feature to avoid entry of unauthorized user.

8. **Manage Information:**

1.1   Through this module works to provide Overview

Digital Voting is a portal that is proposed to automate the functionality of voting. Currently we go polling center for voting but what about physically challenged voters and they faced very difficulty to complete this process so this project also focus this issue.

There are multiple advantages for this portal:

- Transparency.
- Time saving
- Voting from other countries.
- No need of Police /Army in the election duty.
- Normal Transportation facility during voting
- Fraud Detection

<u>Modules:</u>

9. Search Ads:

Through this module Viewer can view ads, can also send response to particular poster.

10. Login :

information details about various companies, institutes, banks, famous places and social activities etc.

11. Poster   :

Through this module a poster can post a new ad, update ad details. He can update including user name and password and search and shortlist the Viewers, check the status of advertisements' response. He can reply to the viewers In short this module deals with the whole advertisement process.

12. Administrator:

This is the Administrator's module by which he keep the eye on whole site and maintain and upgrade the site's service for sake of users. Administrator can show banner ads of others, which help the site in revenue.

The software produced by this project exceeded expectations and would be suitable to be used in industry.

Project management is a key factor of this task to ensure the strict deadlines are adhered to. It is also of paramount importance that tried and tested practices and techniques from the field are adhered to ensure that no common development project mistakes are reproduced.

2. ORGANIZATION OVERVIEW

3. SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Here in the Email to Fax server project, a detailed study of existing system is carried along with all the steps in system analysis. An idea for creating a better project was carried and the next steps were followed.

3.1 Scope of investigation

The aim of this project is to design, build and test a "Gr. Noida Yellow Pages" System. This will be a vastly complex software development project which will take approximately 5 months to complete. The project will be split up into stages and documented thoroughly throughout.

Project management is a key factor of this task to ensure the strict deadlines are adhered to. It is also of paramount importance that tried and tested practices and techniques from the field are adhered to ensure that no common development project mistakes are reproduced.

3.3 Proposed System

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system

helps the user to work user friendly and he can easily do his jobs without time lagging.

3.4.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

3.4.2 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- ➢ Does the existing technology sufficient for the suggested one?
- ➢ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

3.4.3 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- ➢ The costs conduct a full system investigation.
- ➢ The cost of the hardware and software.
- ➢ The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

3.4.4 BEHAVIORAL FEASIBILITY

This includes the following questions:

➢ Is there sufficient support for the users?
➢ Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

4. <u>DESIGN INTRODUCTION</u>

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.1 LOGICAL DESIGN:

The logical flow of a system and define the boundaries of a system. It includes the following steps:

➢ Reviews the current physical system – its data flows, file content, volumes, Frequencies etc.

➢ Prepares output specifications – that is, determines the format, content and Frequency of reports.

➢ Prepares input specifications – format, content and most of the input functions.

➢ Prepares edit, security and control specifications.

➢ Specifies the implementation plan.

➢ Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.

➢ Reviews benefits, costs, target dates and system constraints.

4.2 PHYSICAL DESIGN:

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

➢ Design the physical system.

➢ Specify input and output media.

➢ Design the database and specify backup procedures.

➢ Design physical information flow through the system and a physical design

➢ Plan system implementation.

➢ Prepare a conversion schedule and target date.

➢ Determine training procedures, courses and timetable.

- ➢ Devise a test and implementation plan and specify any new hardware/software.
- ➢ Update benefits , costs , conversion date and system constraints

Design/Specification activities:

Concept formulation.

- ➢ Problem understanding.
- ➢ High level requirements proposals.
- ➢ Feasibility study.
- ➢ Requirements engineering.
- ➢ Architectural design.

4.3 MODULE DESIGN

- ✓ Search Ads
- ✓ Poster
- ✓ Registration
- ✓ Administrator
- ✓ Login

4.4 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The

input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

➢ What data should be given as input?

➢ How the data should be arranged or coded?

➢ The dialog to guide the operating personnel in providing input.

➢ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

➢ Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

➢ It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

➢ When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

4.5 OUTPUT DESIGN

➢ A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate

need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

➢ Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

➢ Select methods for presenting information.

➢ Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

➢ Convey information about past activities, current status or projections of the Future.

➢ Signal important events, opportunities, problems, or warnings.

➢ Trigger an action.

➢ Confirm an action.

4.6 Data Flow Diagram

The Data flow Diagram shows the flow of data. It is generally made of symbols given below:

(1)   A **square** shows the Entity: -

(2)   A **Circle** shows the Process: -

(3)   An **open Ended Rectangle** shows the data store: --

(4)   An **arrow** shows the data flow:-

The DFD can be up to several levels.

 The 0 level DFD states the flow of data in the system as seen from the outward in each module.

The first level DFD show more detail, about the single process of the 0 level DFD

The second level DFD can show even more details and so on.

Context Level DFD

Registration

Voter

Login

Post details

Digital Voting System

Edit & Maintain details

View Responses

Edit Information

View Regd. Users
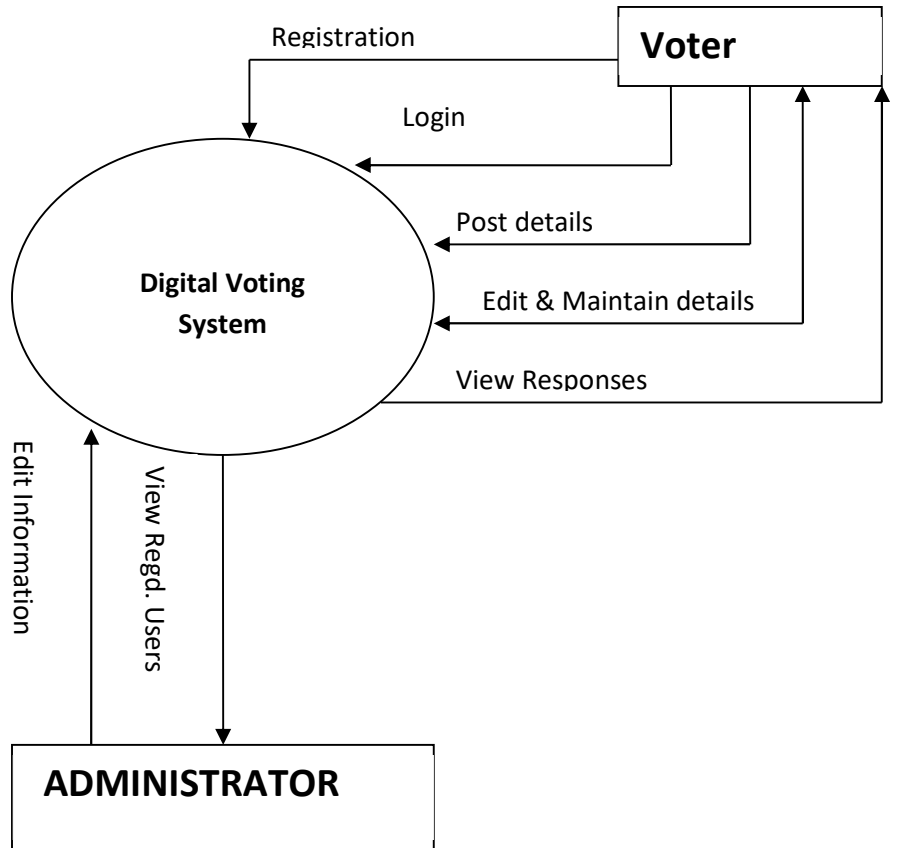
ADMINISTRATOR

First Level DFD (Administrator)

4.7 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design.

The organization of the data in the database is aimed to achieve the following two major objectives.

➢ Data Integrity
➢ Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible.

Normalization is carried out in this system for the following reasons.

➢ To structure the data so that there is no repetition of data , this helps in saving.

- ➢ To permit simple retrieval of data in response to query and report request.
- ➢ To simplify the maintenance of the data through updates, insertions,
- ➢ Deletions.
- ➢ To reduce the need to restructure or reorganize data which new application
- ➢ Requirements arise.

4.7.1 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

RELATIONS, DOMAINS & ATTRIBUTES:

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

RELATIONSHIPS:

- ➢ Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.

- ➢ Entity Integrity enforces that no Primary Key can have null values.
- ➢ Referential Integrity enforces that no Primary Key can have null values.
- ➢ Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.
- ➢ Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

NORMALIZATION:

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These includes:

- ➢ Normalize the data.
- ➢ Choose proper names for the tables and columns.
- ➢ Choose the proper name for the data.

First Normal Form:

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows "relations within relations" or "relations as attribute values within tuples". The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each

table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form:

According to Second Normal Form, for relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that

are fully functionally dependent on it. This step helps in taking out data that is only dependant on apart of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form:

According to Third Normal Form, Relation should not have a nonkey attribute functionally determined by another nonkey attribute or by a set of nonkey attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the nonkey attributes that functionally determines other nonkey attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non key attribute.

4.8 TABLES STRUCTURE

Poster: This table maintains details of poster.

Field Name	Data Type
Poster_id	Varchar2
Username	Varchar2
Password	Varchar2
Email	Varchar2
Address	Varchar2
Category	Varchar2
PhoneNo	Number

Ads: This table maintains details of advertisements.

Field name	Data type
AD id	Varchar2
Category	Varchar2
Title	Varchar2
City	Varchar2
Status	Varchar2
Date	Varchar2
Responses	Varchar2
Description	Varchar2

Category: This table defines the category for ads.

Field Name	Data Type
Category_id	Varchar2
Category_name	Varchar2
Category_description	Varchar2

Administrator: this stores and maintains status of administrator.

Field Name	Data Type
Username	Varchar2
Password	Varchar2

ManageInfo: this store the category of Information and their descriptions.

Field Name	Data Type
Id	Varchar2
Category	Varchar2
Description	Varchar2

5. <u>SOFTWARE ENVIRONMENT</u>

JAVA

Java is a small, simple, safe, object oriented, interpreted or dynamically optimized, byte coded, architectural, garbage collected, multithreaded programming language with a strongly typed exception-handling for writing distributed and dynamicaly extensible programs.

Java is an object oriented programming language. Java is a high-level, third generation language like C, FORTRAN, Small talk, Pearl and many others. You can use java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do.Special programs called applets that can be downloaded from the internet and played safely within a web browser. Java a supports this application and the follow features make it one of the best programming languages.

- ➤ It is simple and object oriented
- ➤ It helps to create user friendly interfaces.
- ➤ It is very dynamic.
- ➤ It supports multithreading.
- ➤ It is platform independent
- ➤ It is highly secure and robust.
- ➤ It supports internet programming

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun's Java platform. The language derives

much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code which can run on any Java virtual machine (JVM) regardless of computer architecture.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun made available most of their Java technologies as free software under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

The Java language was created by James Gosling in June 1991 for use in a set top box project. The language was initially called *Oak*, after an oak tree that stood outside Gosling's office - and also went by the name *Green* - and ended up later being renamed to *Java*, from a list of random words. Gosling's goals were to implement a virtual machine and a language that had a familiar C/C++ style of notation.

Primary goals

There were five primary goals in the creation of the Java language:

1. It should use the object-oriented programming methodology.
2. It should allow the same program to be executed on multiple operating systems.
3. It should contain built-in support for using computer networks.
4. It should be designed to execute code from remote sources securely.
5. It should be easy to use by selecting what were considered the good parts of other object-oriented languages.
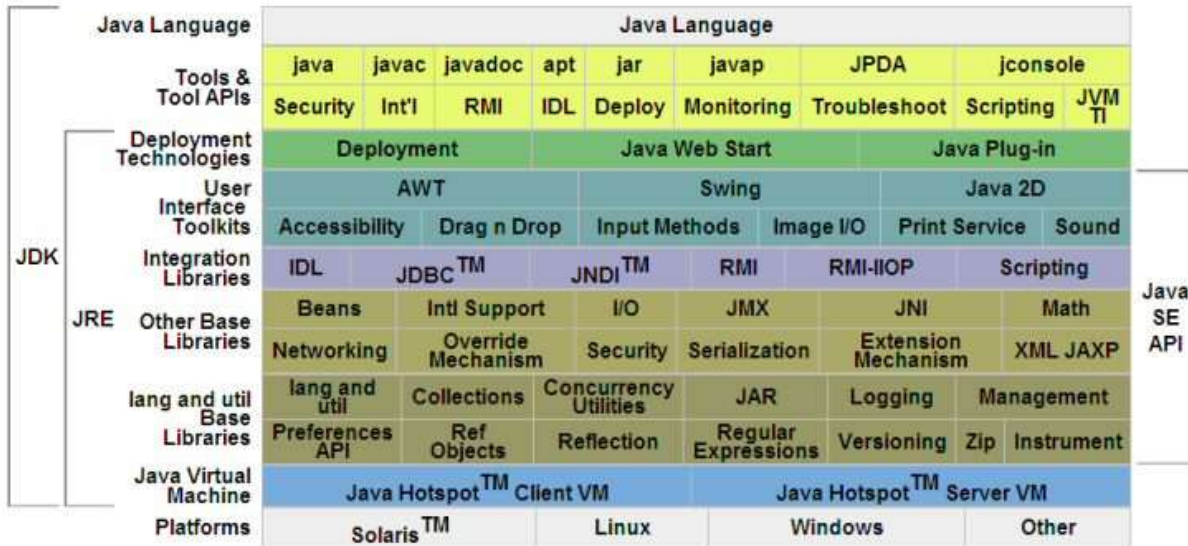
The Java platform is the name for a bundle of related programs, or platform, from Sun which allow for developing and running programs written in the Java programming language. The platform is not specific to any one processor or operating system, but rather an execution engine (called a virtual machine) and a compiler with a set of standard libraries which are implemented for various hardware and operating systems so that Java programs can run identically on all of them.

Different "editions" of the platform are available, including:

➢ Java ME (Micro Edition): Specifies several different sets of libraries (known as profiles) for devices which are sufficiently limited that supplying the full set of Java libraries would take up unacceptably large amounts of storage.
➢ Java SE (Standard Edition): For general purpose use on desktop PCs, servers and similar devices.
➢ Java EE (Enterprise Edition): Java SE plus various APIs useful for multi-tier client-server enterprise applications.

The Java Platform consists of several programs, each of which provides a distinct portion of its overall capabilities. For example, the Java compiler, which converts Java source code into Java bytecode (an intermediate language for the Java Virtual Machine (JVM)), is provided as part of the Java Development Kit (JDK). The sophisticated Java Runtime Environment (JRE), complementing the JVM with a just-in-time (JIT) compiler, converts intermediate bytecode into native machine code on the fly. Also supplied are extensive libraries (pre-compiled into Java bytecode) containing reusable code, as well as numerous ways for Java applications to be deployed, including being embedded in a web page as an applet.There are several other components, some available only in certain editions.

The essential components in the platform are the Java language compiler, the libraries, and the runtime environment in which Java intermediate byte code "executes" according to the rules laid out in the virtual machine specification.

Java Language	Java Language								
Tools & Tool APIs	java	javac	javadoc	apt	jar	javap	JPDA		jconsole
	Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI
Deployment Technologies	Deployment			Java Web Start			Java Plug-in		
User Interface Toolkits	AWT			Swing			Java 2D		
	Accessibility		Drag n Drop		Input Methods	Image I/O	Print Service		Sound
Integration Libraries	IDL	JDBC™		JNDI™		RMI	RMI-IIOP	Scripting	
Other Base Libraries	Beans	Intl Support		I/O	JMX		JNI	Math	
	Networking	Override Mechanism		Security	Serialization		Extension Mechanism	XML JAXP	
lang and util Base Libraries	lang and util	Collections		Concurrency Utilities	JAR		Logging	Management	
	Preferences API	Ref Objects		Reflection	Regular Expressions		Versioning	Zip	Instrument
Java Virtual Machine	Java Hotspot™ Client VM				Java Hotspot™ Server VM				
Platforms	Solaris™		Linux		Windows		Other		

(JDK, JRE, Java SE API annotations appear alongside the diagram)

JAVA VIRTUAL MACHINE

The heart of the Java Platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT compiler within the *Java Virtual Machine*, or JVM. The JIT compiler translates the Java byte code into native processor instructions at run-time and caches the native code in memory during execution.

The use of byte code as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have

"warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter.

Although Java programs are Platform Independent, the code of the Java Virtual Machine (JVM) that execute these programs are not. Every Operating System has its own JVM.

CLASS LIBRARIES

In most modern operating systems, a large body of reusable code is provided to simplify the programmer's job. This code is typically provided as a set of dynamically loadable libraries that applications can call at runtime. Because the Java Platform is not dependent on any specific operating system, applications cannot rely on any of the existing libraries. Instead, the Java Platform provides a comprehensive set of standard class libraries, containing much of the same reusable functions commonly found in modern operating systems.

The Java class libraries serve three purposes within the Java Platform. Like other standard code libraries, they provide the programmer a well-known set of functions to perform common tasks, such as maintaining lists of items or performing complex string parsing. In addition, the class libraries provide an abstract interface to tasks that would normally depend heavily on the hardware and operating system. Tasks such as network access and file access are often heavily dependent on the native capabilities of the platform. The Java java.net and java.io libraries implement the required native code internally, then provide a standard interface for the Java applications to perform those tasks. Finally, when some underlying platform does not support all of the features a Java application expects, the class libraries can either

emulate those features using whatever is available, or at least provide a consistent way to check for the presence of a specific feature.

PLATFORM INDEPENDENCE

One characteristic, platform independence, means that programs written in the Java language must run similarly on any supported hardware/operating-system platform. One should be able to write a program once, compile it once, and run it anywhere.

This is achieved by most Java compilers by compiling the Java language code *halfway* (to Java bytecode) – simplified machine instructions specific to the Java platform. The code is then run on a virtual machine (VM), a program written in native code on the host hardware that interprets and executes generic Java bytecode. (In some JVM versions, bytecode can also be compiled to native code, either before or during program execution,

resulting in faster execution.) Further, standardized libraries are provided to allow access to features of the host machines (such as graphics, threading and networking) in unified ways. Note that, although there is an explicit compiling stage, at some point, the Java bytecode is interpreted or converted to native machine code by the JIT compiler.

The first implementations of the language used an interpreted virtual machine to achieve portability. These implementations produced programs that ran more slowly than programs compiled to native executables, for instance written in C or C++, so the language suffered a reputation for poor performance. More recent JVM

implementations produce programs that run significantly faster than before, using multiple techniques.

One technique, known as *just-in-time compilation* (JIT), translates the Java bytecode into native code at the time that the program is run, which results in a program that executes faster than interpreted code but also incurs compilation overhead during execution. More sophisticated VMs use *dynamic recompilation*, in which the VM can analyze the behavior of the running program and selectively recompile and optimize critical parts of the program. Dynamic recompilation can achieve optimizations superior to static compilation because the dynamic compiler can base optimizations on knowledge about the runtime environment and the set of loaded classes, and can identify the *hot spots* (parts of the program, often inner loops, that take up the most execution time). JIT compilation and dynamic recompilation allow Java programs to take advantage of the speed of native code without losing portability.

Another technique, commonly known as *static compilation*, is to compile directly into native code like a more traditional compiler. Static Java compilers, such as GCJ, translate the Java language code to native object code, removing the intermediate bytecode stage. This achieves good performance compared to interpretation, but at the expense of portability; the output of these compilers can only be run on a single architecture. Some see avoiding the VM in this manner as defeating the point of developing in Java; however it can be useful to provide both a generic bytecode version, as well as an optimised native code version of an application.

AUTOMATIC MEMORY MANAGEMENT

One of the ideas behind Java's automatic memory management model is that programmers be spared the burden of having to perform manual memory management. In some languages the programmer allocates memory for the creation of objects stored on the heap and the responsibility of later deallocating that memory also resides with the programmer. If the programmer forgets to deallocate memory or writes code that fails to do so, a memory leak occurs and the program can consume an arbitrarily large amount of memory. Additionally, if the program attempts to deallocate the region of memory more than once, the result is undefined and the program may become unstable and may crash. Finally, in non garbage collected environments, there is a certain degree of overhead and complexity of user-code to track and finalize allocations. Often developers may box themselves into certain designs to provide reasonable assurances that memory leaks will not occur.

In Java, this potential problem is avoided by automatic garbage collection. The programmer determines when objects are created, and the Java runtime is responsible for managing the object's lifecycle. The program or other objects can reference an object by holding a reference to it (which, from a low-level point of view, is its address on the heap). When no references to an object remain, the Java garbage collector automatically deletes the unreachable object, freeing memory and preventing a memory leak. Memory leaks may still occur if a programmer's code holds a reference to an object that is no longer needed—in other words, they can still occur but at higher conceptual levels.

The use of garbage collection in a language can also affect programming paradigms. If, for example, the developer assumes that the cost of memory allocation/recollection is low, they may choose to more freely construct objects

instead of pre-initializing, holding and reusing them. With the small cost of potential performance penalties (inner-loop construction of large/complex objects), this facilitates thread-isolation(no need to synchronize as different threads work on different object instances) and data-hiding. The use of transient immutable value-objects minimizes side-effect programming.

Comparing Java and C++, it is possible in C++ to implement similar functionality (for example, a memory management model for specific classes can be designed in C++ to improve speed and lower memory fragmentation considerably), with the possible cost of adding comparable runtime overhead to that of Java's garbage collector, and of added development time and application complexity if one favors manual implementation over using an existing third-party library. In Java, garbage collection is built-in and virtually invisible to the developer. That is, developers may have no notion of when garbage collection will take place as it may not necessarily correlate with any actions being explicitly performed by the code they write. Depending on intended application, this can be beneficial or disadvantageous: the programmer is freed from performing low-level tasks, but at the same time loses the option of writing lower level code. Additionally, the garbage collection capability demands some attention to tuning the JVM, as large heaps will cause apparently random stalls in performance.

Java does not support pointer arithmetic as is supported in, for example, C++. This is because the garbage collector may relocate referenced objects, invalidating such pointers. Another reason that Java forbids this is that type safety and security can no longer be guaranteed if arbitrary manipulation of pointers is allowed.

PERFORMANCE

Java's performance has improved substantially since the early versions, and performance of JIT compilers relative to native compilers has in some tests been shown to be quite similar. The performance of the compilers does not necessarily indicate the performance of the compiled code; only careful testing can reveal the true performance issues in any system.

JAVA RUNTIME ENVIRONMENT

The Java Runtime Environment, or *JRE*, is the software required to run any application deployed on the Java Platform. End-users commonly use a JRE in software packages and Web browser plugins. Sun also distributes a superset of the JRE called the Java 2 SDK (more commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

One of the unique advantages of the concept of a runtime engine is that errors (exceptions) should not 'crash' the system. Moreover, in runtime engine environments such as Java there exist tools that attach to the runtime engine and every time that an exception of interest occurs they record debugging information that existed in memory at the time the exception was thrown (stack and heap values). These Automated Exception Handling tools provide 'root-cause' information for exceptions in Java programs that run in production, testing or development environments.

➤ **Java 2 Enterprise Edition (J2EE)**

The J2EE platform uses a multitier distributed application model. Application logic is divided into components according to function, and the various application components that make up a J2EE application are installed on different machines depending on the tier in the multitier J2EE environment to which the application

component belongs. Figure 1-1 shows two multitier J2EE applications divided into the tiers described in the following list. The J2EE application parts shown in Figure 1-1 are presented in J2EE Components.

- ➢ Client-tier components run on the client machine.
- ➢ Web-tier components run on the J2EE server.
- ➢ Business-tier components run on the J2EE server.
- ➢ Enterprise information system (EIS)-tier software runs on the EIS server.

Although a J2EE application can consist of the three or four tiers shown in Figure 1-1, J2EE multitier applications are generally considered to be three-tiered applications because they are distributed over three different locations: client machines, the J2EE server machine, and the database or legacy machines at the back end. Three-tiered applications that run in this way extend the standard two-tiered client and server model by placing a multithreaded application server between the client application and back-end storage.
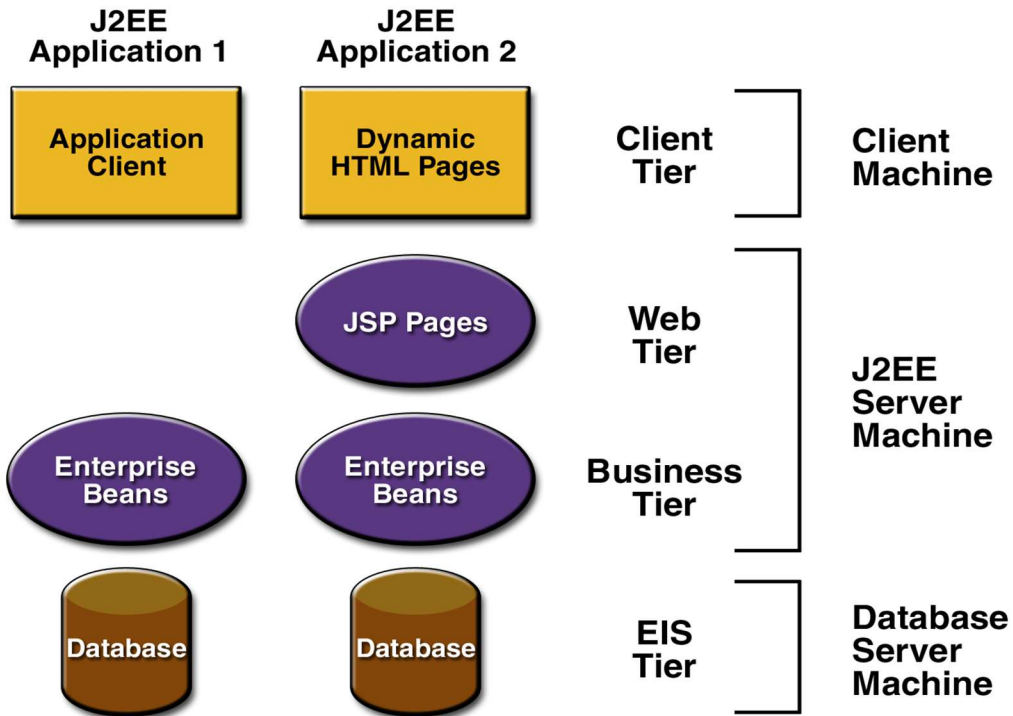
Figure 1-1 Multitier Applications

J2EE COMPONENTS

J2EE applications are made up of components. A *J2EE component* is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and that communicates with other components. The J2EE specification defines the following J2EE components:

- ➢ Application clients and applets are components that run on the client.
- ➢ Java Server and Java Server Pages (JSP) technology components are Web components that run on the server.
- ➢ Enterprise JavaBeans (EJB) components (enterprise beans) are business components that run on the server.

J2EE components are written in the Java programming language and are compiled in the same way as any program in the language. The difference between J2EE components and "standard" Java classes is that J2EE components are assembled into a J2EE application, verified to be well formed and in compliance with the J2EE specification, and deployed to production, where they are run and managed by the J2EE server.

J2EE CLIENTS

A J2EE client can be a Web client or an application client.

WEB CLIENTS

A Web client consists of two parts: dynamic Web pages containing various types of markup language (HTML, XML, and so on), which are generated by Web components running in the Web tier, and a Web browser, which renders the pages received from the server.

A Web client is sometimes called a *thin client*. Thin clients usually do not do things like query databases, execute complex business rules, or connect to legacy applications. When you use a thin client, heavyweight operations like these are off-loaded to enterprise beans executing on the J2EE server where they can leverage the security, speed, services, and reliability of J2EE server-side technologies.

APPLETS

A Web page received from the Web tier can include an embedded applet. An applet is a small client application written in the Java programming language that executes in the Java virtual machine installed in the Web browser. However, client systems will likely need the Java Plug-in and possibly a security policy file in order for the applet to successfully execute in the Web browser.

Web components are the preferred API for creating a Web client program because no plug-ins or security policy files are needed on the client systems. Also, Web components enable cleaner and more modular application design because they provide a way to separate applications programming from Web page design. Personnel involved in Web page design thus do not need to understand Java programming language syntax to do their jobs.

APPLICATION CLIENT

A J2EE application client runs on a client machine and provides a way for users to handle tasks that require a richer user interface than can be provided by a markup language. It typically has a graphical user interface (GUI) created from Swing or Abstract Window Toolkit (AWT) APIs, but a command-line interface is certainly possible.

Application clients directly access enterprise beans running in the business tier. However, if application requirements warrant it, a J2EE application client can open an HTTP connection to establish communication with a servlet running in the Web tier.

JAVABEANS COMPONENT ARCHITECTURE

The server and client tiers might also include components based on the JavaBeans component architecture (JavaBeans component) to manage the data flow between an application client or applet and components running on the J2EE server or between server components and a database. JavaBeans components are not considered J2EE components by the J2EE specification.

JavaBeans components have instance variables and get and set methods for accessing the data in the instance variables. JavaBeans components used in this way are typically simple in design and implementation, but should conform to the naming and design conventions outlined in the JavaBeans component architecture.

J2EE SERVER COMMUNICATION

The client communicates with the business tier running on the J2EE server either directly or, as in the case of a client running in a browser, by going through JSP pages or servlets running in the Web tier.J2EE application uses a thin browser-based client or thick application client. In deciding which one to use, you should be aware of the trade-offs between keeping functionality on the client and close to the user (thick client) and off-loading as much functionality as possible to the server (thin client). The more functionality you off-load to the server, the easier it is to distribute, deploy, and manage the application; however, keeping more functionality on the client can make for a better perceived user experience.
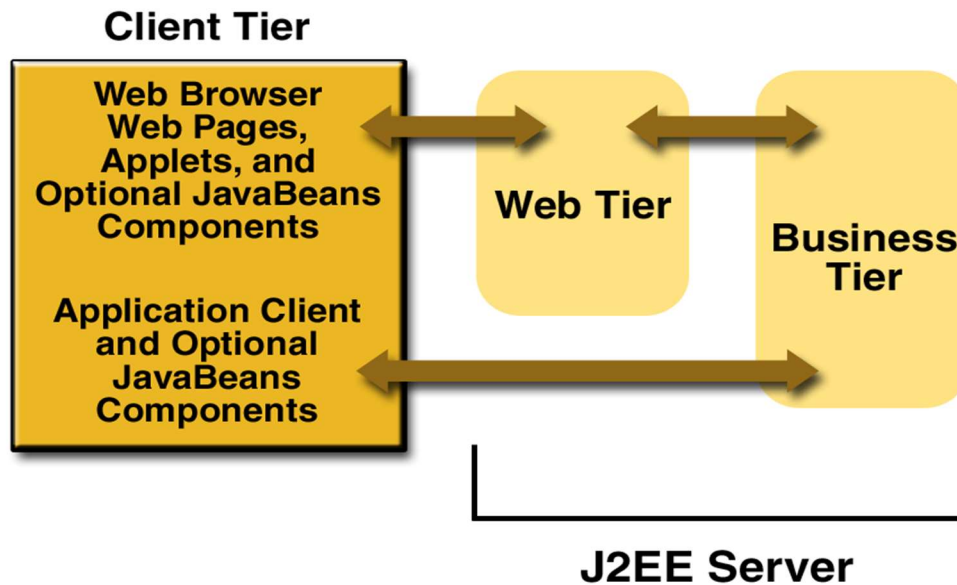
Figure 1-2 Server Communications

WEB COMPONENT

J2EE Web components can be either servlets or JSP pages. *Servlets* are Java programming language classes that dynamically process requests and construct responses. *JSP pages* are text-based documents that execute as servlets but allow a more natural approach to creating static content.Static HTML pages and applets are bundled with Web components during application assembly, but are not considered Web components by the J2EE specification. Server-side utility classes can also be bundled with Web components and, like HTML pages, are not considered Web components.Like the client tier and as shown in , the Web tier might

include a JavaBeans component to manage the user input and send that input to enterprise beans running in the business tier for processing.
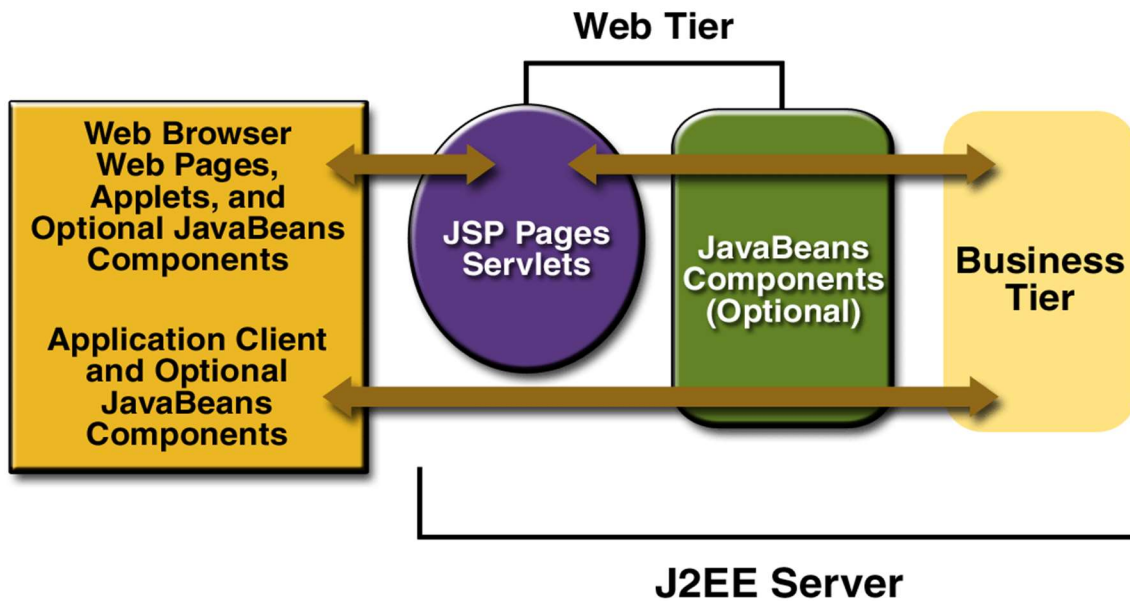
Figure 1-3 Web Tier and J2EE Application

BUSINESS COMPONENT

Business code, which is logic that solves or meets the needs of a particular business domain such as banking, retail, or finance, is handled by enterprise beans running in the business tier. An enterprise bean also retrieves data from storage, processes it (if necessary), and sends it back to the client program.

Figure 1-4 Business and EIS Tiers
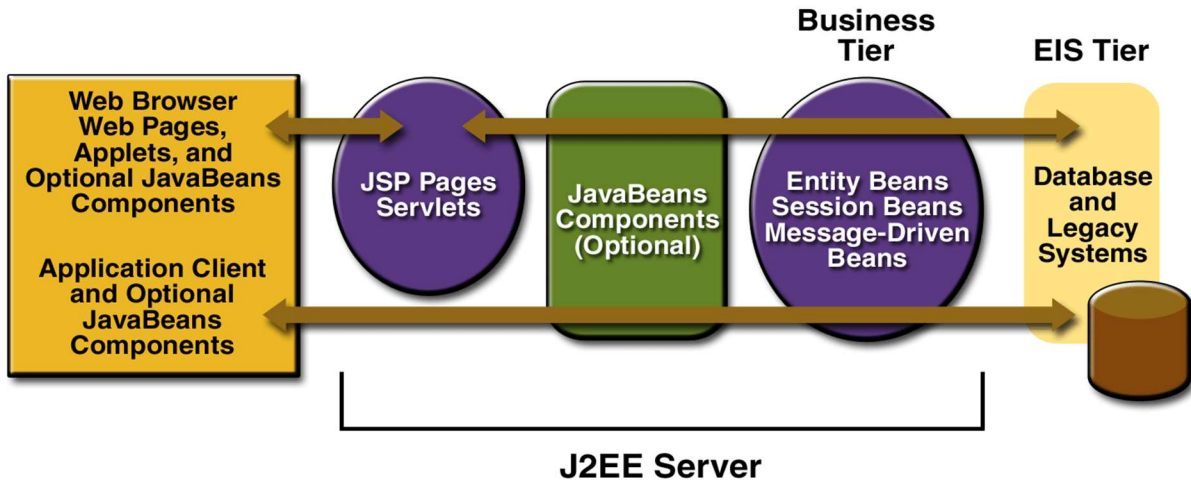
1. **Figure 1-4 Business and EIS Tiers**

There are three kinds of enterprise beans: session beans, entity beans, and message-driven beans. A *session bean* represents a transient conversation with a client. When the

client finishes executing, the session bean and its data are gone. In contrast, an *entity bean* represents persistent data stored in one row of a database table. If the client terminates or if the server shuts down, the underlying services ensure that the entity bean data is saved.

A *message-driven bean* combines features of a session bean and a Java Message Service (JMS) message listener, allowing a business component to receive JMS messages asynchronously. This tutorial describes entity beans and session beans.

ENTERPRISE INFORMATION SYSTEM TIER

The enterprise information system tier handles enterprise information system software and includes enterprise infrastructure systems such as enterprise resource planning (ERP), mainframe transaction processing, database systems, and other legacy information systems. J2EE application components might need access to enterprise information systems for database connectivity

JAVA SERVER PAGE (JSP)

Java Server Pages technology is the Java platform technology for building applications containing dynamic Web content such as HTML, DHTML and XML. The Java Server Pages technology enables the authoring of Web pages that create dynamic content easily but with maximum power and flexibility.

The Java Server Pages technology offers a number of advantages:

➤ **Write Once, Run Anywhere properties:**
The Java Server Pages technology is platform independent, both in its dynamic Web pages, its Web servers, and its underlying server components. You can author JSP pages on any platform, run them on any Web server or Web enabled application server, and access them from any Web browser. You can also build the server components on any platform and run them on any server.

➤ **High quality tool support**

The Write Once, Run Anywhere properties of JSP allows the user to choose *best-of-*

breed tools. Additionally, an explicit goal of the Java Server Pages design is to

enable the creation of high quality portable tools.

➢ □*Reuse of components and tag libraries*

The Java Server Pages technology emphasizes the use of reusable components such

as: JavaBeans components, Enterprise JavaBeans components and tag libraries.

These components can be used in interactive tools for component development and

page composition. This saves considerable development time while giving the cross-

platform power and flexibility of the Java programming language and other scripting

languages.

➢ **Separation of dynamic and static content**

The Java Server Pages technology enables the separation of static content from dynamic content that is inserted into the static template. This greatly simplifies the creation of content. This separation is supported by beans specifically designed for the interaction with server-side objects.

➢ **Support for scripting and actions**

The Java Server Pages technology supports scripting elements as well as actions.

Actions permit the encapsulation of useful functionality in a convenient form that can also be manipulated by tools; scripts provide a mechanism to glue together this functionality in a per-page manner.

JSP ARCHITECTURE

JSPs are built on top of SUN's servlet technology. JSPs are essential an HTML page with special JSP tags embedded. These JSP tags can contain Java code. The JSP file extension is .jsp rather than .htm or .html. The JSP engine parses the .jsp and creates a Java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the JSP is probably slower the first time it is accessed. Any time after this the special compiled servlet is executed and is therefore returns faster.

JAVA SCRIPT

JavaScript is a programming language that allows scripting of events, objects, and actions to create Internet applications. A website development environment that will allow the creation of Interactive Web Pages. The coding techniques capable of accepting a client's requests and processing these requests. The web site development environment should also provide the facility for 'validating' user input. With JavaScript, forms are a consideration in nearly every page you design. Capturing user requests is traditionally done via a 'form'. So the web site needs to have facilities to create forms. Text fields and textareas can dynamically change in response to user responses.

TOMCAT 5.0

Tomcat is a servlet container and Java Server Pages implementation it may be used stand alone ,or in conjunction with several popular web servers .

➢ Apache version 1.3 or later

➢ MS Internet Information Server ,version 4.0 or later

➢ MS personel web server, version 4.0 or later

➢ NetScape enterprise server , version 3.0 or later

Tomcat is a security update release.This release closes a whole that potentially allowed access to resourse protected by a <security constraint > in web.xml.

Installing and Running Tomcat 5.0

Tomcat requires a Java Runtime Environment (JRE).Conformant to JRE 1.1 or later including any Java2 platform system.If one wishes to develop applications you will need a java compiler , such as the one included in a java development kit 1.1 or later environment including JDKs conformant with Java2.

FEATURES OF OS

This project work is done on the windows 2000 professional, which is the operating system. An operating system is a set of software tools designed to make it easy for people or programmers to make optimum use of the computer. People who use computers have different levels of needs and interest. These peoples can be separated can be two groups, users and programmers. The user wants a convenient set of commands to manage files of

data or programs, copy and run application package while a programmer used as a set of tools that can be held together and debug programs.

No matter where you are working, your computer will be easier to use and manage, because Microsoft Windows 2000 Professional is more compatible and more powerful than any workstation you've used before. The main features of Windows 2000 Professional operating system are

- ➢ Easier to use.
- ➢ Easier to manage
- ➢ More compatible
- ➢ More powerful

EASIER TO USE

With Windows 2000 Professional, you have faster access to information, and you arte able to accomplish tasks more quickly and easily.

Windows 2000 Professional makes it easier to:

- ➢ Work with files.
- ➢ Find information.
- ➢ Personalize your computing environment.
- ➢ Work on the web.
- ➢ Work remotely

EASIER TO MANAGE

You and your network administrators can work more efficiently now, because many of the most common computer-management tasks are automated are streamlined with Windows 2000 Professional.

With Windows 2000, your workstation will be easier to:

- ➢ Set up.
- ➢ Administrator
- ➢ Support.

MORE COMPATIBLE

Windows 2000 Professional offers increased compatibility with different types of network and with a wide array of legacy hardware and software.

Windows 2000 also provides:

- • Improved driver support.
- • Increased support for new-generation hardware multimedia technologies.

MORE POWERFUL

For all your computing needs, Windows 2000 Professional provides:

Industrial-strength reliability

- ➢ The highest level of security
- ➢ Powerful performance

Windows 2000 also contains the following features:

PORTABILITY

- ➢ Windows file protection protects core system files from being overwritten by application installs.
- ➢ Driver certification provides safeguards to assure you that device drivers have not been tampered with and reduces your risk of installing non-certified drivers.

- Full 32 bit operating system minimizes the chance of application failures and unplanned reboots.

MOBILITY

- Hibernate turns off your computer and monitors after a predetermined time while retaining your desktop on disk.
- Offline viewing makes entire WebPages with graphics available for viewing offline
- Synchronization manager allows you to compare and update your offline files and folders with those on the network.
- Smart battery gives you a more accurate view of your battery's life enabling you to reduce power to specify functions to extend your battery power.
- Hot docking tells you dock or undock your notebook computer without changing hardware configuration or rebooting.
- Universal Serial Bus (USB) lets you connect and disconnect a wide array of peripherals such as joysticks, scanners and camcorders without configuring or rebooting your computer.
- J2EE 1394 provides a higher band width connection for devices that require faster data transfer.

MAINTAINABILITY

- System preparation tool (sys prep) helps administrators clone computer configuration systems and applications.
- Set up manager provides a graphical wizard that guides administrators in designing installation scripts.
- Multilingual support allows users to easily create, read and edit documentation in hundreds of languages.
- Windows 2000 server offers 25% faster performance than Windows 95 or Windows 98 on systems with 64MB or more of memory.
- 32 bit architecture allows you to run more programs and perform more faster at the same time than Windows 95 or 98.
- Windows 2000 can support to 4GB of Ram and two symmetric multiprocessors.
- Encrypting file system (EFS) encrypts each file with a randomly generated key.
- IP Security (IP Sec) support protected data transmitted across a network.
- Kerberos support provides industry standard high-strength authentication with a fast, single login to windows 2000 enterprise resources.

INTERNET CAPABILITY

- Internet Information Services (IIS) 5.0 includes web and FTP server support, as well as support for Front-page transactions, Active Server Pages (ASP) and database connections.
- Windows 2000 has strong development platform support for dynamic HTML behaviors and XML.
- Intelliforms alleviates the tedious of filling out forms on the web by automatically entering your name, address or other information that you have securely stored on your computer.

➤ Automated proxy automatically locates a proxy server configures Internet Explorer 5.0 to connect to the internet through the server.

5.1 SOFTWARE AND HARDWARE SPECIFICATION

<u>Hardware Specification:</u>

Microprocessor	: - Pentium-4 class processor, 450
megahertz (MHz)	
RAM	: - 256 MB of RAM
Hard Disk	: - 40 gigabytes (GB) on installation drive

<u>Software Specification:</u>

- Windows 98 / XP /7 or any operating system

- Oracle10g as Database Sever

- JDK 5 or above

- Weblogic8.1 or above/ Apache Tomcat 5.0 or above

5.2   Project Architecture and Platform

Front End   **:**   **Html, JavaScript, CSS**

- User friendly GUI
- Client Side Validation

Middle-Tier   **:**   **J2ee with MVC Architecture**

Model   **:** JavaBeans

View   **:** Servlet, Jsp

Controller **:** Filter

Back End   **:**   **Oracle10gXE**

- Security
- Performance
- Scalability
- Reliability
- Support RDMS concepts

5.2.1 CODING STANDARDS

The standard used in the development of the system is Microsoft Programming standards. it includes naming conversions of variables, constants and objects, standardized formats or labelling and commenting code, spacing, formatting and indenting.

5.2.2 NAMING CONVENTION

Classes' names and interface names will start with capital letter. The function names will start with small letters and the first letter of each word in the function name will be in capital letter.

5.2.3 LABELS AND COMMENTS

Sufficient labels and comments are included in the description of it for the benefits if the developer and other programmers who might examine it later.

User Interface

- For all the entry screen frames are used which will show the type of the user who is currently logged in and the menus.
- Standard actions are used for standard actions.
- Same font is related properties are used for similar screens.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the users are trained in the new procedures.

6. TESTING INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified?. Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation **:** Are we doing the right job?

Verification **:** Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis

looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives.

They are

- Testing is a process of executing a program with the intend of findingan error.
- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncovered errors in the software also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

6.1 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

6.1.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that

information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

6.1.2 INTEGRATION TESTING

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing in Sell-Soft System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

6.1.3 VALIDATION TESTING OR SYSTEM TESTING

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

6.1.4 OUTPUT TESTING OR USER ACCEPTANCE TESTING

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points

- ➢ Input Screen Designs,
- ➢ Output Screen Designs,
- ➢ Online message to guide the user and the like.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Validation Checking:

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software test-validation checks may begin. Validation can be defined in many ways, but a simple definition (Albeit Harsh) is that validation succeeds when software functions in a manner that can be reasonably expected by a customer. Software validation is achieved through a series of black-box tests to be conducted and a test procedure defines specific test cases that will be used in attempt to uncover errors in conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied; all performance requirements are achieved; documentation is correct and human –Engineered and other requirements are met. Once the application was made free of all logical and interface errors , inputting dummy data to ensure that the software developed satisfied all the requirements of the user did validation checks .However , the data are created with the intent of determining whether the system will process them correctly .

In the proposed system, if the clients click the send button after selecting a file from his file list, then the system will show the confirmation message for sending files. Similarly if a client makes an attempt to download a file from the server file list, then also the system will show the confirmation message for downloading. This is how the data validations were made in the proposed system.

7. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a

successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

Implementation is the stage of the project where the theoretical design is tuned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- ♦ Careful planning.

- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.

7.1 Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

7.1.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more

important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

7.1.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered.

It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

7.1.3 Operational Document

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. Education involves right atmosphere and motivating the user. A documentation providing the whole operations of the system is being developed in such a way that the user can work with it in well consistent way. The system is developed user friendly so that the user can work the system from the tips given in the application itself. Useful tip and guidance is given inside the application itself to help the user. Users have to be made aware that what can be achieved with the new system and how it increases the performance of the system. The user of the system should be given a general idea of the system before he uses the system.
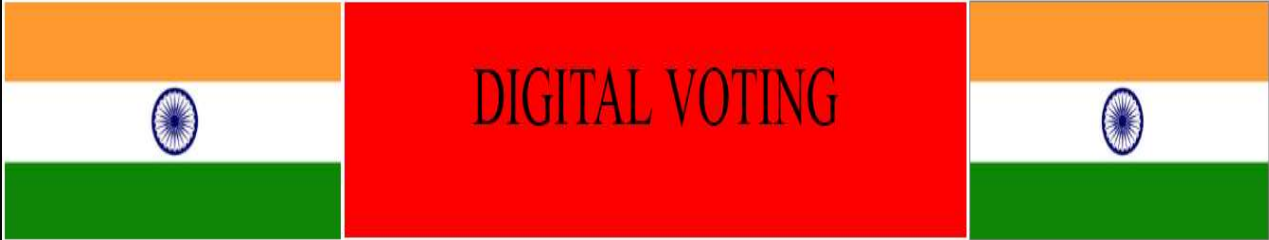
7.1.4 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes". Maintenance may be defined by describing four activities that are undertaken after a program is released for use

8. SCREEN SHOTS
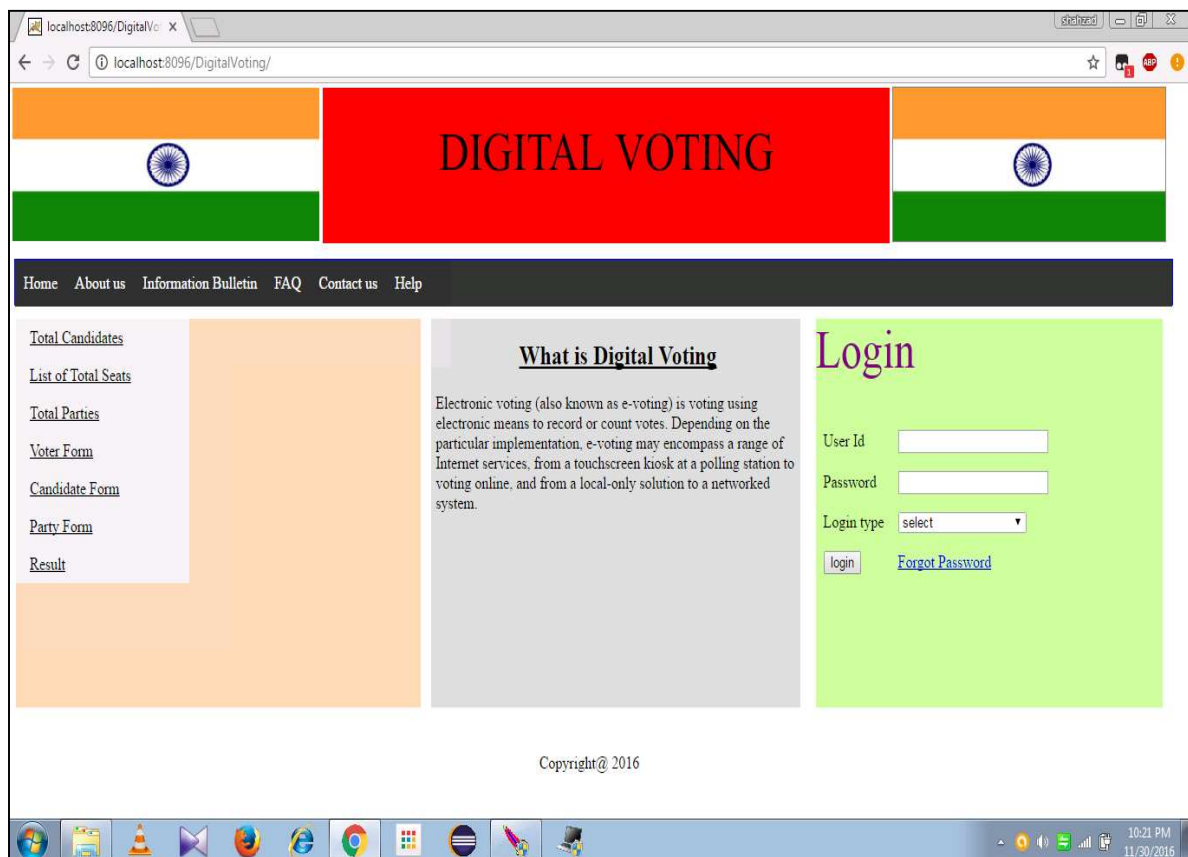
← → C | ⓘ localhost:8096/DigitalVoting/ec_login.jsp

DIGITAL VOTING

Election Commission of India

Welcome:india

Logout

Create UserId

Total Candidates

List of Total Seats

Create Seat

Total Parties

Approve Party

Approve Candidate

Result

What is Election Commission

The Election Commission of India is an autonomous constitutional authority responsible for administering election processes to Lok Sabha, Rajya Sabha, state legislatures and the offices of the President and Vice President in India.[1] Election Commission operates under the authority of Constitution, and subsequently enacted Representation of the People Act. The Supreme Court of India has held that where the enacted laws are silent or make insufficient provision to deal with a given situation in the conduct of elections, the Election Commission has the residuary powers under the Constitution to act in an appropriate manner.

Copyright@2016

localhost:8096/DigitalVoting/create_party.jsp

DIGITAL VOTING

Election Commission of India

Home

Total Candidates

List of Total Seats

Create Seat

Total Parties

Voter Form

Candidate Form

Party Form

Result

Party Name

Party Abbreviation

submit

Copyright@2016

localhost:8096/DigitalVoting/create_seats.jsp
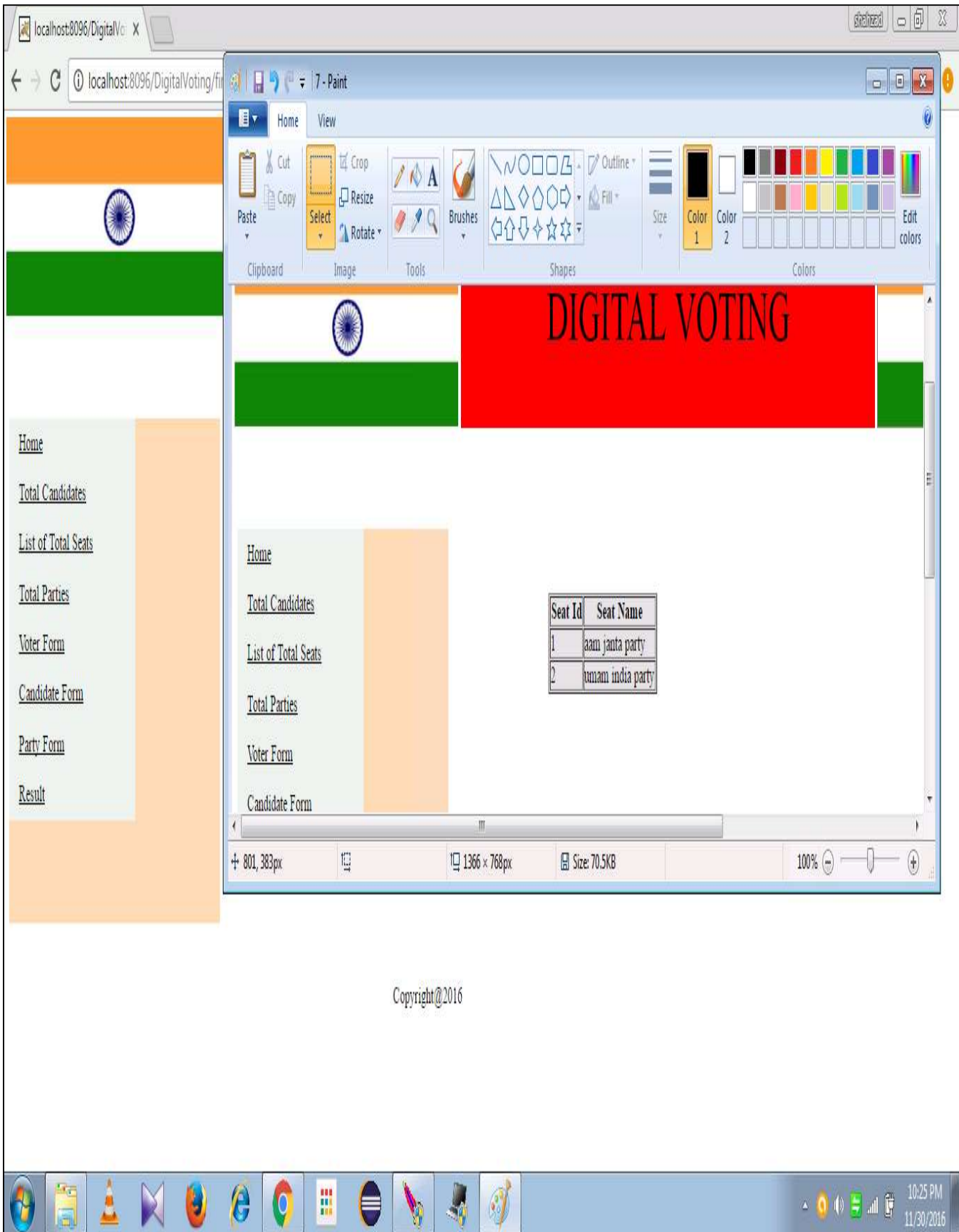
DIGITAL VOTING

Election Commission of India

Logout

Welcome:null

- Home
- Total Candidates
- List of Total Seats
- Total Parties
- Voter Form
- Candidate Form
- Party Form
- Result

Seat Name [   ]

submit

Copyright@2016

← → C ① localhost:8096/DigitalVoting/create_seats.jsp

DIGITAL VOTING

Election Commission of India

Logout

Welcome:null

Home

Total Candidates

List of Total Seats

Total Parties

Voter Form

Candidate Form

Party Form

Result

Seat Name [   ]

submit

Copyright@2016

DIGITAL VOTING

Election Commission of India

Home

Total Candidates

List of Total Seats

Create Seat

Total Parties

Voter Form

Candidate Form

Party Form

Result

Candidate Name

Select Seat   krishna nagar ▼

Select Party   aam janta party ▼

submit

Copyright@2016

DIGITAL VOTING

Home

Total Candidates

List of Total Seats

Total Parties

Voter Form

Candidate Form

Party Form

Result

Seat Id	Seat Name
1	aam janta party
2	nmam india party

Copyright@2016

Home

Total Candidates

List of Total Seats

Total Parties

Voter Form

Candidate Form

Party Form

Result

DIGITAL VOTING

Home

Total Candidates

List of Total Seats

Total Parties

Voter Form

Candidate Form

Seat Id	Seat Name
1	aam janta party
2	umam india party

Copyright@2016

localhost:8096/DigitalVoting/create_userid.jsp

DIGITAL VOTING

Election Commission of India

Logout

Welcome:india

Create UserId

Total Candidates

List of Total Seats

Create Seat

Total Parties

Approve Party

Approve Candidate

Result

Total UserIds:10

Generate UserId Automatically

Copyright@2016

localhost:8096/DigitalVoting/total_seats.jsp

DIGITAL VOTING

Election Commission of India

Welcome:india

Logout

Create UserId

Total Candidates

List of Total Seats

Create Seat

Total Parties

Approve Party

Approve Candidate

Result

Seat Id	Seat Name	Action
1	krishna nagar	edit delete
2	tugalpur	edit delete

Copyright@2016

9. CONCLUSION

The project titled as "Gr. Noida Yellow Pages" is a web based application. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

9.1 Scope for Future Enhancement

In future we can able to add more modules in this project GP-Desk. In the proposed system the user is provided with a choice of data screen, which are similar in formats to the source documents. Data entry errors can be minimized through validity checks. After the verification only the data are placed in the permanent database. The software can be developed further to include a lot of modules because the proposed system is developed on the view of future, for example we should develop the system as a database independent using JDBC so we can connect it to any other database. Now the proposed system is based on

PC and intranet but in the future if we need to convert it into internet then we need to change the front end only because we are developing this on the basis of OOP technology and most of the business logic's are bounded in the class files and module like reusable components.

10. BIBILOGRAPHY

BOOKS:

- Charles Hampfed (2000) 'Instant Java Server Pages' University of Toronto
- Herbert Schildt (2000) 'Java Complete Reference' Tata McGraw Hill
- John Zukowski (2000) 'Mastering Java2' BPB Publications
- Jamie Jaworsky 'J2EE Bible' Techmedia
- Stefen Denninger 'Enterprise Java Beans-2.1' Author's Press
- Ian Somerville 'Software engineering'
- Rajeev mall 'Software engineering'
- Elmasri Navathe 'Fundamentals of database systems'

ONLINE REFERENCE:

- www.theserverside.com
- www.java.sun.com