Recruitment System

PROJECT WORK DONE AT

HCL INFOSYSTEMS LTD
CHENNAI-17

PROJECT REPORT

P- 567

Submitted in partial fulfillment of the
Requirements for the award of the degree of
Master of Computer Application
Of Bharathiar University, Coimbatore.

Submitted By

K.KABILAN
(Reg. No. 9838M0505)

Guided By

EXTERNAL GUIDE
N.S.GANESH
G.M-BANKING
HCL INFOSYSTEMS LTD
Chennai-17

INTERNAL GUIDE
S.ANDREWS M.Sc., PGDPM
LECTURER
Computer Science & Engineering
Kumaraguru college of technology
Coimbatore-06

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641 006.

CERTIFICATE

This is to certify that the project work entitled

"Recruitment System"

Done by

K.KABILAN
(REG. NO. 9838M0505)

Submitted in partial fulfillment of the requirements for the award of the degree of **Master Of Computer Application** of **Bharathiar University.**

[illegible] 26/4/01 [illegible] 26/04/2001

Professor and Head **Internal Guide**

Submitted to University Examination held on 11/05/2001

[illegible] 11/05/2K1 [illegible] 11/5/01

Internal Examiner **External Examiner**

HCL INFOSYSTEMS LTD.

Professional Services Organisation
Software Technology Park
"Thapar House" II Floor
43/44, Montieth Road, Chennai - 600 008
Ph : 8553450 / 8511293 / 8511954
Fax : 8511986

Chennai, April 28 , 2001

TO WHOMSOEVER IT MAY CONCERN

This is to certify that K. Kabilan, student of final year M.C.A., Kumaraguru College of Technology, Coimbatore, completed a project titled "**Recruitment System**" under my guidance. This was done in partial fulfillment of the requirements for award of the said degree. The project was done during the period Dec-2000 to Apr-2001.

His conduct during the period of project work has been exemplary.

I wish him all success in his endeavours.

For HCL INFOSYSTEMS LTD.,

[illegible]

(N. S. Ganesh)
General Manager , Banking and Finance

evolve

Registered Office : 806 - 808 Siddharth, 96 Nehru Place, New Delhi 110 019 | HCL

DECLARATION

I hereby declare that the project entitled

"Recruitment System"

Submitted to **Bharathiar University** as the project work of **Master Of Computer Application** Degree, is a record of original work done by me under the supervision and guidance of **N.S.GANESH General Manager,HCL INFOSYSTEMS LTD, Chennai** and **Mr S.ANDREWS M.Sc,PGDPM Lecturer ,Department of Computer Science and Engneering, Kumaraguru College of Technology, Coimbatore** and this project work has not found the basis for the award of any Degree/Diploma/ Associate ship/Fellowship or similar title to any candidate of any university.

[illegible]

Place: Signature of the student

Date:

ACKNOWLEDGEMENT

I would like to thank the following people who helped me in accomplishing the task of completing this project and writing this report.

Its my great privilege and pleasure of mine to express my deep sense of gratitude to esteemed **Dr. K.K.Padmanaban B.Sc., (Engg)., M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology, Coimbatore.

I am grateful to **Prof.Dr.S.Thangaswamy** Head of the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore for their kindness, which has enabled this project to be a complete one

I express my deep sense of gratitude and indebtedness to my Internal project guide, **Mr. S.ANDREWS, M.Sc., PGDPM,** Lecturer, Department of Computer Science and Engineering, KUMARAGURU COLLEGE OF TECHNOLOGY for the useful and excellence guidance throughout my effort.

I thank **Mr.N.S.GANESH** General Manager, Banking, HCL INFOSYSTEM LTD Chennai, who as a guide gave me excellent guidance, valuable suggestions and guided me through out the project.

At the outset I would to thank **Mr. NISHAR AHMED** Associate Consultant HCL INFOSYSTEM LTD, Chennai, for his precious guidance, suggestions and continuous help, which have been valuable assets in the progress of this project.

I reciprocate the kindness shown to me by my friends and family members, staff members of my department of Computer Science.

Synopsis:

This Project is developed to customize the complete activity of HCL INFOSYSTEMS LTD. This project has the following components.

i Administrator

i.i Question Bank Repository Manager
i.ii Examination Controller
i.iii Reports
i.iv candidates profile

ii Examinee

i Administrator:

Administrator is control the overall system that is splitted into following four categories.

i.i Question Bank Repository Manager:

Question Bank repository manager actually stores the Questions. Here, administrator can add Questions, change an existing Question. Their question type classifies the questions. They are grouped based on the subject. So new subjects for certain objects can be added.

i.ii Examination Controller

This is a very secure system. Here the system allows the Examiner In-charge to select Question Papers. It allows the Examiner In-charge to publish the examination date (date range), assigning the candidate to specific question paper, and also giving the date (date range) within which the user must take the exam.

i.iii Reports:

The administrator can generate the following reports.

Candidates details:

Here the candidate's personal information along with the score in which he/she has taken the exam.

Subject wise details:

It publishes the results with user-id, user name, and score for corresponding subject within the specified date (date range).

i.v candidates profile:

Candidates profile stores the details of the candidates who are all enter eligible to the company conditions.

ii Examinee:

Here the candidate logs in and takes the exam that has been assigned.

INTRODUCTION

1.1Project overview:

This project will facilitate the HR department in its recruitment process. Recruitment System is a project intended to be used for maintaining the candidates details and to conduct the online exam. Based upon the score the candidates are short listed for the interview. The system provides option to assign the Question Paper to be answered by the examinee within a specified time. The system provides enough security for confidentiality of the Question Papers.

1.2Organization profile:

HCL

HCL INFOSYSTEMS LTD

HCL Infosystems Ltd (HCL Insys) is India's premier information technology company. With its in-depth expertise in developing solutions spanning diverse technologies, HCL Insys aims to propel its course on to the high-growth path of 'Total Technology Integration'. Leveraging its expertise in total technology solutions and services, HCL Insys offers value-added services in key areas such as SAP implementation, software applications, networking consultancy and management and a range of support services.

Our Strategy

To capture the two ends of the market spectrum - enterprise solutions and PCs - HCL Insys has made significant strategic infrastructure investments in the Professional Services Organization (PSO), the Support Services Organization (SSO), and in its manufacturing plants at Pondicherry. The approach is to view hardware as one of the key components of the total solution. The build-up of the services business (both of PSO and SSO) will enable HCL Insys to offer complete solutions and will also raise manufacturing volumes to internationally competitive levels.

Professional Services Organization (PSO)

Established in 1994, HCL Insys' PSO provides single-window enterprise solutions in key vertical segments of telecommunications, manufacturing, finance & banking, government, utilities and transportation.

Spearheading HCL Insys' thrust on software exports, the PSO's infrastructure comprises a taskforce of 600 software specialists, four software factories, and dedicated 'Centers of Excellence' for SAP and telecom solutions. Significant SAP implementation projects by the PSO include General Motors, HM Lancer, Kalyani Brakes and Samsung. PSO's projects involve high-level IT consulting, large systems integration projects and functional consulting and implementation services for ERP (Enterprise Resource Planning).

Its recent software export projects include a complete end-to-end aviation insurance project for the US-based CIGNA Property & Casualty, one of the largest insurance providers in the world; a manufacturing execution systems project for Consilium; a Year 2000 project for RCC Denmark and a module for telecom billing software for Saville of The US.

Support Services Organization (SSO)

Continuously increasing customer expectations and the application of its focus on integrated enterprise solutions have strengthened the HCL Insys SSO's capabilities in supporting installation types ranging from single to large, multi-location orders. The SSO, comprising a direct support force of over 800 members, is operational at 143 locations across the country and is the largest such human resource of its kind in the IT business. A majority of the SSO has been specially trained in supporting solutions, the Company's key focus area.

Adhering to Stringent Quality Methods

HCL Insys' manufacturing facilities at Pondicherry is equipped with a capacity of over 15000 machines per month and adhere to stringent quality standards and global processes. With the largest installed PC base in the country, three indigenously developed and manufactured PC brands - the 'Infiniti', 'Busybee' and the 'Beanstalk' and its robust manufacturing facilities, HCL Insys aims to further leverage its dominance in the PC market. Also manufactured at the Pondicherry facility, HCL Insys' 'Infiniti' line of business computing products is incorporated with leading edge products from world leaders such as Intel. A fully integrated and business-ready intranet family of servers and workstations, the 'Infiniti' line is targeted at medium and large companies to help them to Manage their intranet-related applications.

As customers' needs mature, the IT industry has witnessed a continual evolution through a progressively finer segmentation of markets. To become a dominant player in providing global IT services, HCL Insys has reorganized and consolidated its hardware and services businesses. Towards this, it is setting up overseas subsidiaries in the US, the U.K., Singapore and Malaysia and has acquired the assets of HCL Infosolutions and HCL Peripherals, and the customer support activities, related products and human resources of HCL Office Automation.

Aims

HCL Insys aims to comprehensively address total technology solutions in the services and hardware sectors of the IT industry in the new millennium.

HCL Insys' initiatives in setting national and international industry standards in technology, solutions and processes led to the establishment of two unique programmers - "Program Infiniti" in 1992 and "Enterprise 2000" in 1994. "Program Infiniti" is a consistent and continuous plan designed to upgrade and modernize HCL Insys' entire hardware and software capability to world-class standards. "Enterprise 2000" is a unique programmed with an aim to provide IT solutions to customers, to place them at the cutting edge of their respective businesses by the year 2000 and beyond. In keeping with its objective of total quality management, over 70 per cent of its employees are being trained under the Phil Crosby methodology of Quality Education System.

Complete Range of Internet Solutions

Following the government's decision to open up the distribution of Internet services to the private sector, HCL Insys, under a new initiative called WWW.OW, will provide a complete range of solutions for the Internet. These services include setting up infrastructure for ISPs, solutions for cybercafes, information kiosks, and Net-on-TV, corporate intranet and extranet, infrastructure for smart cities and a set of management services such as facilities management and network management.

Strategic Alliances

In 1992, HCL formed a joint venture company, HCL HP, with the international computer giant, Hewlett-Packard (HP), and precipitated a technological leap by achieving world class manufacturing expertise in the country for HP's RISC/UNIX based business servers and workstations. In 1997, HCL and its joint venture partner, HP, reorganized their joint business activities in India through the buying back of HP's 26 per cent of equity in HCL Hp by the HCL promoters.

With an enlarged business focus, HCL Insys has strengthened its multiple strategic alliances with specialists to include world leaders such as Intel for PCs and PC Servers; Microsoft, Novell and SCO for operating systems and software solutions; Toshiba Corp. for business automation equipment; SAP AG for specialist ERP solutions; and Oracle, Sybase and Informix for RDBMS platforms.

SYSTEM STUDY AND ANALYSIS

System Study and Analysis:

System analysis is the processing of gathering and interpreting facts, diagnosing problems and the information to recommend to improvements to the system. System analysis and design can be defined as the process of analyzing a business organization or system, assessing the requirement and designing the procedures, which the use of computers. First step in system analysis is the initial investigation to determine whether the report is valid and feasible before a recommendation is made us to do nothing improve or modify the existing system or build a new system. At the heart system analysis is a detailed understanding of all the important facts of the business areas under investigation.

The main task involved in it is

Learn the details of the current system.

Develop insight into future demands.

Document the details of the current system.

Evaluation the effectiveness and efficiency of the current system.

Recommend necessary revision and improvements.

Document the new system features.

Fact Finding Techniques

The first job was to find out what was the present system and how it works. For that purpose I met the user of the system in corporate department. In meeting arranged they told me their problems with the existing system and also explained all the details about the corporate procedures and how it is done. So from this we found a good response for changing the existing system.

After this in order to get an overview of the existing system and to find out the user problems with the existing system and design a counterpart for the same, it was very necessary t develop a deep understanding during the operation, It was further essential to understand the outputs generated by the system the inputs to the system and various codes and tests to implemented during the operation. All this was achieved through the various fact-finding methods described below.

Meeting and discussions:

This is the first step towards fact finding. We had a brief discussions with the project manager who wanted the existing system to be converted into intranet so as to provide a better system which overcome all the difficulties faced by the existing system. Lengthy discussions were held with the various users of the system during first few days and there after meetings were held at regular intervals with the management staffs. This helped in manufacturing the existing system in details and to plan for the proposed system.

Reviewing records:

Various files and records were supplied which gave exact picture f all documents, format and reports, which should be generated by the proposed system. It also gave the idea of the volume of data and frequency of various operations.

On the spot observation:

This was one of the important methods of fact-finding. On the spot observation helped us to notice many minute details to be incorporated in the system while the user key in the information details. The areas were the user find it difficult during the process and the places were he want some modification to be done were taken into account, so that the requirement of the user is given first priority.

Feasibility Study:

One of the important outcomes of the preliminary investigations is the determination that the system requested is feasible.

Feasibility study is a major, who contributes to the analysis of the system, the decision of the system's analyst whether to design a particular system or not depends on its feasibility study, therefore the feasibility forms the very base of the system.

Technical feasibility:

The hardware and software needed for designing and implementation of the system is already available. Hence the system is found to be technically feasible.

Economic feasibility:

As the hardware and software required is already available, the only cost involved is that of coding, implementing and maintaining the system. Hence the system is found to be economically feasible.

Operational feasibility:

The user will be given practice training in using the system. The new system will be user friendly. Thus there will be no resistance from the user. Hence the system is operationally feasible.

Motivational feasibility:

Since computerization will lead to reduction of tremendous manual interaction, the user feels motivated to work in the new system. It simplifies the job of the hr-manager of the organization.

2.1Existing system:

Since the earlier system is done manually to select the candidates, conduct the exam and store the details of the employees. As a result of that it takes more time.

2.2Proposed system:

Objective:

This project intends to provide the complete activity of recruitment system.

Features:

Gives details about candidates and their career details.

Online exam.

Gives detail reports about the exam.

Stores details of the candidates who attend the exam.

2.3 User Characteristics:

The users of this package are HR department and the candidates. Context sensitive help facility to the users.

General Constraints:

This subsection provides a general description of any other items that will limit the future developer's option for designing the system. Regulatory policy: All the deliverables in the life cycle of this system should confirm to the existing standards and guidelines for the development of software products.

Operating Environment:

Recruitment system has to be a client-server application using the INTRANET in HCL with JavaScript, HTML as the front-end and ORACLE as the backend server.

Availability:

The application will be available so long as the server machine is on and the Java Web Server or 'Servletrunner' program is running in it.

PROGRAMMING ENVIRONMENT

3.Programming environment

Hardware configuration:

Windows NT operating system 4.0:

From its inception, Microsoft Windows NT was designed to be a robust, portable operating system that would be maintainable, flexible, and secure over time. This describes the design decisions that were made during the initial planning of Windows NT; describes each of the design goals in priority order (as documented in the original design documentation, "NT OS/2 Subsystem Design Rationale," [June 1989]).

The Mission:

When the Windows NT development team was formed in 1989, it had a clear mission: to design and build a personal computer operating system that would meet the current and future operating system needs of the PC platform. To meet this objective, the design team identified the following market requirements:

To provide easy portability to other 32-bit architectures

To provide scalability and multiprocessing support

- To support distributed computing, allowing multiple computers to share resources
- To support the application programming interfaces (APIs) required by the Portable Operating System Interface for UNIX (POSIX)

- To provide U.S. government Class 2 (C2) security features, and to provide a path to Class B1 and beyond

The Design Goals:

Based on market requirements and Microsoft's development strategy, the original Microsoft NT design team established a set of prioritized goals. Note that from the outset, the priority design objectives of Windows NT were *robustness* and *extensibility*:

1. **Robustness.** The operating system must actively protect itself from internal malfunction and external damage (whether accidental or deliberate), and must respond predictably to software and hardware errors. The system must be straightforward in its architecture and coding practices, and interfaces and behavior must be well specified.

2. **Extensibility and maintainability.** Windows NT must be designed with the future in mind. It must grow to meet the future needs of original equipment manufacturers (OEMs) and of Microsoft. And the system must be designed for maintainability—it must accommodate changes and additions to the API sets it supports and the APIs should not employ flags or other devices that drastically alter their functionality.

3. **Portability.** The system architecture must be able to function on a number of Platforms with minimal recoding.

4. **Performance.** Algorithms and data structures that lead to a high level of performance and that provide the flexibility needed to achieve our other goals must be incorporated into the design.

Microsoft Windows NT 4.0 has enhanced performance, supports the GUI used to Windows 95, and includes system policy editor, much like the Windows 95 version, that allows for granular management of client workstation desktops.

Windows NT 4.0 also integrates Internet services; with peer web services on Windows NT workstations and fully features web, Gopher, and FTP services in the server version. A network protocol, Point-to-Point Tunneling Protocol (PPTP), supports RAS connectivity via the Internet Windows NT 4.0 gives hints of the upcoming object-oriented version of Windows NT, which Microsoft has referred to as Cairo.

3.1Hardware Requirements:

Server with MS Windows NT Server 4.0

Pentium II 266Hz (or Higher)
RAM 128MB (or Higher)
Sufficient Disk Space (~ 200MB for the application)
CD ROM Drive
Network Access
Monitor that supports 256/more colors

Client with MS Windows 95/98/NT Workstation 4.0

Pentium 200Hz (or Higher)

RAM 64MB (or Higher)

Sufficient Disk Space (~ 100MB for the application)

CD ROM Drive

Network Access

Monitor that supports 256/more colors

Features of Windows NT operating system:

Reliable:

NT is made up of series of separate subsystems; so one program, for example, can't crash the whole system. NT also supports fault-tolerant devices like redundant disk arrays.

Secure:

Both users and processed must have permissions to access resources. All access is logged the system.

Multithreaded and symmetric-Processing:

NT can handle many " trains of thought" simultaneously. If your system has more than one CPU, NT will distribute tasks among all the available processors. It's fast!

Client/server:

NT server supports a wide range of network protocols and RPC (Remote Procedure Call) communications, making it easy for programmers to create distributed applications.

Ease of use:

For all its power and complexity, NT is easy to set up and maintain. Most administrative function can be performed with the powerful graphical tools provided.

Wide Range of Hardware Support:

NT was designed from "Day one" to be a portable operating system. Thus days, you can run NT server on Intel, DEC, Alpha, MIPS, and PowerPC computers, ranging from a simple 486 desktop all the way up to a 28-processors monster computer. You can find the system that meets your needs.

Pre-emptive Multitasking:

NT can switch among multiple program "preemptively", which means that NT can force the switch to occur without waiting for the program to relinquish control of the system.

Multiple Message Queues:

With graphics windowing user interface systems, a "message" is any action that the program needs to know about; you pressed a key moved the mouse or spoke a voice command.

Protected Memory:

Under NT, a program can't screw up the whole system when it crashes.

Security:

There is extensive security support in NT. All you need to do to gain access to the local computer under Windows 95 is hit cancel at the windows login screen. Os/2 wrap desktop security falls in between (again). You can't control access to individual files on the local computer as finely with OS/2 as you can with NT.

3.2Description of software and tools used:

3.2.1Networks:

Introduction to Networks – Connecting people not just computers:

A network is an interconnection of two or more computers for the purpose of sharing information (data, schedule, e-mail) and resources (printers, storage devices, and applications). In simpler terms, a network is a two or more computer attached with a communication medium (cable), where another can use information and hardware on one machine.

Uses Of Computer Networks:

Resource Sharing: The primary goal of networks is resource sharing i.e. to make all programs, data and equipment available to anyone on the network without regard to the physical distance between the resource and the user.

High Reliability: Having alternate sources of supply can provide reliability. In addition, the presence of multiple CPUs means that if one of them goes down, the others maybe able to take over its work.

Saving Money: Since physical resources in a network can be shared between many computers on the network, a lot of money is saved which would have otherwise gone into providing separate resources for every terminal.

Powerful Communication Medium: A computer network can provide a powerful communication medium among widely separated people. Thus human-to-human interaction is being encouraged.

Language of Networks:
LAN (Local Area Network) describes a network whose geographic span is less than one mile.

WAN (Wide Area Network) describes a network whose geographic span is greater than one mile. Often, multiple LANs are attached to create a WAN.

Node is a general term for any device on a network. Every client, server, hub, and gateway can be called a node.

Network bandwidth, load, overhead, and lag are terms associated with the capability of a network to transmit data over it's communication medium.

Bandwidth: The bandwidth of a network is the amount of data that can be transmitted over a given segment of communication media within a specific time, usually measured in bits, Kilobits, or Megabits per second (bps, Kbps, or Mbps).

Load: The load of a network is a general term used to describe the demands of a network.

Overhead: A network's overhead is the amount of bandwidth required to successfully complete a given networking operation, such as transferring a file or maintaining an active connection to a workstation.

Firewall: A hardware or software barrier (or both) that protects a network from unwanted intruders.

Protocol: In networking and communications, the formal specification that define the procedures to follow when transmitting and receiving data. Protocols define the format, timing, sequence, and error checking used on the network.

The OSI Reference Model:

This model is based on a proposal developed by ISO as a first step towards standardizing networks. It is called Open Systems Interface model because it deals with connecting open systems- that is, systems that are open for connection with other systems.

The OSI model has seven layers:

1) Application Layer
2) Presentation Layer
3) Session Layer
4) Transport Layer
5) Network layer
6) Data Link layer
7) Physical layer

Types Of Networks:

Peer-to-peer:

Every machine attached to a peer-to-peer network has the same access rights as every other machine on the network; no centralized location exists for applications. It is a collection of computers that share information equally, where no one machine is the center of the network.

Client-Server:

A client/server network is a collection of computers (servers) that hold shareable resources and computers (clients) that access these resources from the servers. Basically servers provide services. This type of network is designed over the convention that application and data are stored on one or more servers that users can access from any of the clients. Both the client and server can process information, and a client can perform tasks completely independent of the server.

A server can be centralized or dedicated. A centralized server is the single server on a network that handles all the server-specific networking tasks. A dedicated server is a single server on a multi-server network that handles one or more server-specific tasks, for example we can have a printer server, a fax server, a data file server etc.

Intranets:

Intranet Definition

An Intranet is a network based on TCP/IP protocols belonging to an organization, usually a corporation, accessible only by the organization's members, employees or others with authorization. An Intranet's web sites look and act like any other web sites, but the firewall surrounding an Intranet fends off unauthorized access. Like the Internet itself, Intranets are used to share information. Secure Intranets are now the fastest growing segment of the Internet because they are much less expensive to build and manage, than private networks based on proprietary protocols. Therefore, in short, an Intranet exploits Internet technologies to share information and resources in an efficient and dynamic manner. This makes Intranets popular among corporate offices, R&D organizations, educational institutions and the like.

Essential Services Over Intranets:

Intranets have evolved to support certain core services like:

E-mail Messaging System
Directory Services
File and Print Services
Web publishing and Conversion
Newsgroup
Network Management
Search and Retrieval Services
Interactive Support Services
Firewall services
Proxy services

These services will make the corporate environment self sufficient, provide ease of use with universal clients and moreover will produce cost effective solutions.

Intranet Applications:

In a corporate like environment, the Intranet applications will make a viable impact and play a major role in the network information flow. The following are the key areas where Intranet applications make an impact:

Human Resource Management
Sales and Marketing
Product Development
Customer Service and support
Information Publishing
Education and Training
Financial Services
Personal etc

Basic Building Blocks of an Intranet:

The following are the basic components that go into the building of an Intranet:

A Network:

A working Local Area Network is the transport medium of the Intranet. The network servers, as well as the clients could be running any operating system as long as they all support the TCP/IP protocol.

TCP/IP:

Transmission Control Protocol/Internet Protocol is the universal standard for network communication over the Internet and by extension for any Intranet. TCP/IP must be installed and configured on all servers and nodes that will be part of the Intranet.

Hardware:

Hardware comprises of at least one server machine dedicated to hosting Intranet services.

3.2.2Software:

Web Browser:

The Web Browser is a software application used to locate and display Web pages. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer. Both of these are graphical browsers, which means that they can display graphics as well as text. In addition, most modern browsers can present multimedia information, including sound and video, though they require plug-ins for some formats. The browser is heading towards becoming a universal front end for all services in an Intranet. For example IE4 merges access to the Intranet, the Internet and the local hard disk under one common Web interface.

Server Software:

A server is a computer or device on a network that manages network resources. For example, a file server is a computer and storage device is dedicated to storing files. Any user on the network can store files on the server. A print server is a computer that manages one or more printers and a network server is a computer that manages network traffic. A database server is a computer system that processes database queries.

Servers are often dedicated, meaning that they perform no other tasks besides their server tasks. On multiprocessing operating systems, however a single computer can execute several programs at once .A server in this case could refer to the program that is managing the resources rather than the entire computer.

Web Server:

A Web server is nothing but an HTTP server that serves the Web pages available in its directories, in response to browser client requests after authentication based on certain rules defined at server end .It runs on top of the TCP/IP protocol. Every Web server has an IP address and possibly a domain name. For example if the client enters the URL http://www.sun.com/index.html in his browser, this sends a request to the server whose domain is sun.com. The server then fetches the page named index.html and sends it to the browser.

Any computer can be turned into a web server by installing server software and connecting the machine to the Internet/Intranet. There are many Web server software applications, including public domain software from NCSA and Apache and commercial packages from Microsoft, Netscape and others.

Mail Server:

Since e-mail services are usually the major application on Intranets, a server that is dedicated to managing the mail system is necessary. Mail server software based on the SMTP (Simple Mail Transfer Protocol) should be installed on the server machine.

Proxy Server:

A proxy server is software installed on machines that dials-up to the Internet. When the machine connects to the Internet, other users on the network access the Internet through the proxy server, as if each of them had a live connection through a separate modem and a phone line. It is a dramatically effective way to share an Internet connection among many users. It is a server between a client application, such as a web browser and a real server. It intercepts all requests to the real server to see if it can fulfill the requests itself. If not it forwards the requests to the real server. Proxy servers can also be used to filter requests. For example, a company might use a proxy server to prevent its employees from accessing a specific set of web sites.

Firewall:

A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, specially Intranets. All messages entering or leaving the Intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. There are several types of firewall techniques:

Packet Filter:

Looks at each packet entering or leaving the network and accepts it or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition it is susceptible to IP spoofing.

Application Gateway

Applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but can impose performance degradation.

Circuit level gateway

Applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

Proxy server

Intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.

In practice, many firewalls use two or more of these techniques in concert. A firewall is considered a first line of defense in protecting private information. For greater security data can be encrypted.

3.2.3 HTML – HYPER TEXT MARKUP LANGUAGE:

Introduction to HTML:

PERL, CGI, Servlet are useful tools in processing interactive output. But they primarily function in the background. The front-end interface is provided by HTML. HTML is a text manipulation language. HTML is an acronym for Hypertext Markup Language, which is the language that all web browsers use to format web pages on the screen. It is a collection of platform independent styles that defines various components of the WWW documents. Tim Berners-Lee at CERN, the European Laboratory for particle physics in Geneva invented this.

HTML documents are plain-text ASCII files that can be created using any text editor .It is a set of special codes that you embed in text to add formatting and linking information. HTML is based on SGML (Standardized General Markup Language).

SGML - The basis of HTML:

Abbreviation of Standard Generalized Markup Language, it is a system for organizing and tagging elements of a document. SGML was developed and standardized by the International Organization for Standards (ISO) in 1986. SGML itself does not specify any particular formatting; rather it specifies the rules for tagging elements. These tags can then be interpreted to format elements in different ways.

SGML is used widely to manage large documents that are subject to frequent revisions and need to be printed in different formats. Because it is a large and complex system, It is not yet widely used on personal computers. However, the growth of Internet and specially the World Wide Web is creating renewed interest in SGML because the World Wide Web uses the HTML, which is derived from SGML.

The Hypertext Markup Language began as a simple set of text markup commands. It constantly evolved to become the de facto standard for Web authoring. The IETF HTML working group sets HTML standards, which is the official body of research, debate and approval for the HTML standard. The current in use is HTML 4.0.

HTML Documents and Web Pages:

So what kind of documents are in those web site folders? Actually these documents are not much different from the standard word-processing documents. The biggest difference between an everyday run-of-the-mill word-processing document and a document designed for the web is the use of hypertext links. Hypertext links are those little hot spots that you can click on to go from one document to the next. It also takes time to send documents over the Internet. If you put, say 20 pages on your web site, you would not want to download all of them at once because it would take too long for a 20-page document to get there. Instead just have a small home page that contains links to the other documents.]

Home Pages:

The first page that the reader sees when connecting to the web site is called the home page. The purpose of the home page is to give the reader an idea as to what the site is all about, as well as what it has to offer. The home page typically contains links to other documents in the site.

3.2.4 JAVA:

A Brief History of Java:

In 1990,Sun Microsystems began a project called "Green" to develop software for use in consumer electronics. James Gosling, a veteran of classic network software design, was assigned to the new project.

Gosling's solution to this problem was a new language called Oak. Oak preserved the familiar syntax of C++ but omitted the potentially dangerous features like explicit resource references, pointer arithmetic and operator overloading. Oak incorporated memory management directly into the language, freeing the programmer to concentrate on the tasks to be performed by the program. In order to be successful as an embedded system programming language, Oak needed to be able to respond to real-world events within microseconds. It also needed to be portable; that is, it had to be able to run on a number of different microprocessor chips and environments. This hardware independence would allow a toaster manufacturer to change the chip used to run the toaster without changing the software. The manufacturer could also use some of the same code that runs the toaster to run similar equipment, such as a toaster oven.

As Oak matured, the World Wide Web was in a period of dramatic growth, and the development team at Sun realized that Oak was perfectly suited to Internet Programming. In 1994, they completed work on a product known as Web Runner, an early Web Viewer written in Oak. Web Runner was later renamed Hot Java, and it demonstrated the power of Oak as an Internet development tool. Finally, in 1995, Oak was renamed Java (for marketing purposes) and announced at Sun World 95. Since then Java's popularity has been meteoric. Java is now considered an industry standard for Internet development.

The Java programming Language:

The developers at Sun Microsystems describe Java this way:

Java: A simple, object-oriented, distributed, and interpreted, robust, secure, architecturally neutral, portable, high-performance, multithreaded, dynamic language.

Simple: One of the design goals behind Java was to make it familiar to a large number of programmers; hence, it's strong resemblance to C and C++. The designers have simplified matters by removing some of the C++ constructs, such as pointers, that traditionally cause problems for programmers.

Object-oriented: Java is purely object-oriented, unlike C++ where it is not strictly implemented. This means that the programmer can concentrate on the data in the application rather than thinking in terms of procedures.

Distributed: Java is designed to support networking and network operations right from the start.

Interpreted: Java is an interpreted language rather than a language, which you can compile and then run. A Java program can run on any system for which there is a Java interpreter.

Robust: Java is a strongly typed language that allows for comprehensive runtime checking, and with no pointers to worry about, there is no possibility of overwriting some distant memory area and corrupting data.

Secure: Java begins with the assumption that it can trust no one, and it implements several security measures to protect you from attempts to create viruses or invade your file system.

Architecturally neutral: Java does not care about the underlying operating system or hardware. Java byte codes are platform independent, and can run on any machine with a JVM in it.

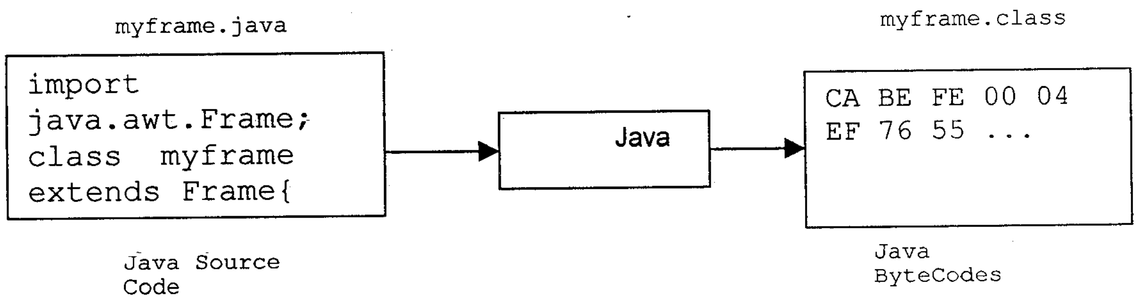
High performance: Just-in-time compilers help Java achieve higher performance than the Java interpreter alone van reach.

Multithreaded: Java supports multiple threads of execution to handle different tasks.

Dynamic: Java was designed to adapt to a changing environment and can load classes as they are needed, even across a network

How java works:

As with many other programming languages, Java uses a compiler to convert human-readable source code into executable programs. Traditional compilers produce code that can be executed by specific hardware; for example, a Windows 95 C++ compiler creates executable programs that work with Intel x86 compatible processors. In contrast the Java compiler generates architecture-independent byte codes. Only a Java Virtual Machine (JVM) can execute the byte codes, which is an idealized Java processor chip implemented in software rather than hardware. The compilation process is illustrated in the figure below.


myframe.java
import java.awt.Frame; class myframe extends Frame{
Java Source Code
Java
myframe.class
CA BE FE 00 04 EF 76 55 ...
Java ByteCodes

Java byte code files are called class files because they contain a single Java class. In simple words, a class represents a group of related routines or an extended data type.

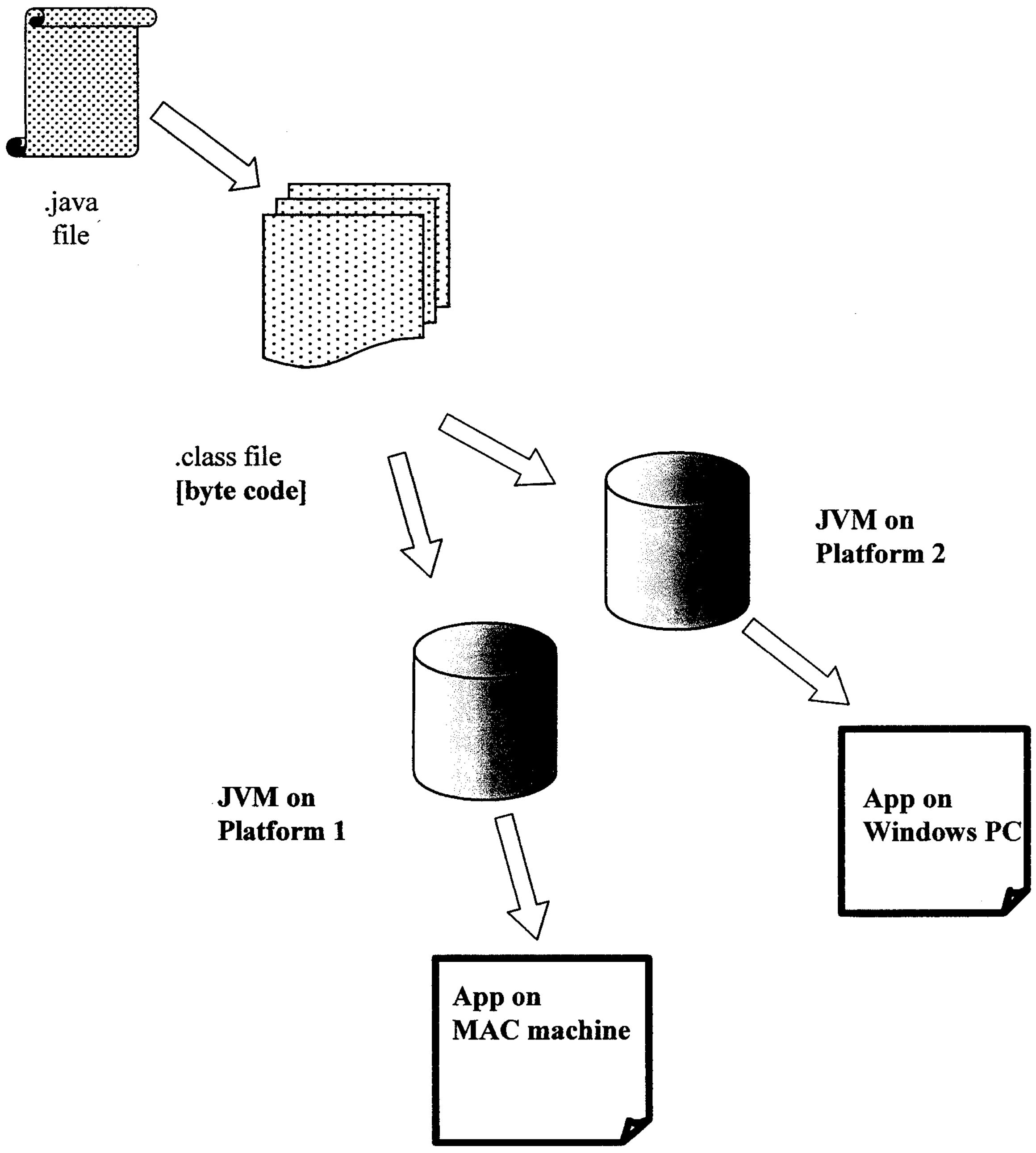
To execute Java byte codes, the VM uses a class loader to fetch the byte codes from a disk or from the network. Each class file is fed to a byte code verifier that ensures that the class is formatted correctly and that the class will not corrupt memory when it is executed. The byte code verification phase adds to the time it takes to load a class, but it actually allows the program to run faster because the class verification is performed only once, not continuously as the program runs.

The execution unit of the VM carries out the instructions specified in the byte codes. The simplest execution unit is an interpreter, which is program that reads the byte codes, interprets their meaning, and then performs the associated function. Interpreters are generally much slower than native code compilers because they continuously need to look up the meaning of each byte code during execution. Fortunately, there is an elegant alternative to interpreting code, called Just-In-Time (JIT) compilation.

The JIT compiler converts the byte codes to native code instructions on the user's machine immediately before execution. Traditional native code compilers run on the developer's machine, are used by programmers, and produce non-portable executables. JIT compilers run on the users machine and are transparent to the user. Figure 4 overleaf illustrates the way JIT compilers work. In the example, both a Macintosh and a Windows PC receive identical byte codes, and each client performs a local JIT compilation.

Java Enabled Browsers:

A Java-enabled Web browser contains it's own VM. Web documents that have embedded Java applets must specify the location of the main applet class file. The Web browser then starts the VM and passes the location of the applet class file to the class loader. Each class file knows the names of any additional class files that it requires. These additional class files may come from the network or from the client machine. This may require the class loader to make a number of additional class-loading operations before the applet starts.



Java - Write Once, Run Anywhere

JVM:

What is a Java Virtual Machine?

At the most elementary level, the Java VM is a software CPU with the Java byte code as the instruction set. A developer writes a Java program and stores the program in a .java file. The Java compiler converts this Java program in the .java file into a .class file consisting of byte codes. The byte codes are the instruction set for the Java VM. The byte codes are of the form:
<opcode><...Parameters>

The instruction code, or opcode, is one byte long (hence the name) and can have many parameters. Currently there are about 220 byte code instructions defined in the Java VM specification.

Like normal program loaders, the Java VM starts the execution of a program by loading the .class file from either the network or local storage. If the program passes the byte code verification phase, the Java VM runtime interpreter reads the byte codes, translates them to the specific operating system/ hardware instructions, and executes them in the target CPU.

To run the Java byte codes, a hardware manufacturer or operating system vendor implements the Java VM on it's hardware/ operating system combination. The Java VM uses the host operating system facilities for memory functions, file system, display, mouse, keyboard, network, other device drivers, processing of threads, and so on.

JVM Performance:

Due to the additional Java VM layer over the host operating system (compared to traditional compiled languages like C and C++), the current Java VMs are considered to be slower than native compiled code. The reasons for the slower performance of Java are:

- **The Verification Phase:**

  The verification phase takes time. As a class is read in, it is verified during runtime, whereas traditional compilers perform verification processes during the compilation of the program. Also, as part of the class is executed, the Java byte codes are converted to native machine instructions. If the same section is executed multiple times, it is converted to native instructions multiple times. It does not cache previously converted byte codes. Additionally, as each instruction is executed, there are multiple safety checks being performed.

- **Instruction Size:**

  Java instructions are all byte-sized codes. Since most operations require objects that are bigger than a byte (an integer is 32 bits), multiple byte codes would need to be read to get the various operators and their parameters.

- **Java as a Stack Machine:**

  Java is implemented as a stack machine, which means that the instructions take the parameters from the operand stack and return the results to the stack. This is due to the fact that the Java VM does not assume ant particular CPU architecture, including the word size and register combinations.

  Normally, compiled programs use very fast register operations, and the parameters needed for an operation are available in the CPU registers most, if not all, of the time. Because registers are in the CPU whereas a stack is in the main memory, register operations are faster than stack operations. Also, traditional compilers provide many types of code optimization where the results of operations are registers and subsequent operations access the results of previous operations from registers.

- **Garbage Collection:**

  The system performs automatic garbage collection for the users, which inevitably impacts performance since everything must stop when the garbage collector runs.

JIT Compiler:

Java VM is much slower than traditional compiled languages. The JIT compiler is a technique that greatly improves performance. The JIT compiler translates the byte code to native machine code just before execution, so you get a byte code compiler instead of an interpreter. It provides portability without sacrificing speed.

3.2.5 JAVA SCRIPT:

JavaScript – Extending HTML:

HTML is a very limited document formatting language. It is based on tags, which instruct the browser how to display a chunk of text or an image. As such, the HTML is limited to a static, one-way interaction with the user. Interaction cannot be static-it requires constructs such as if statements and for loops, which are not a part of the HTML syntax.

These missing constructs are found in JavaScript. This object-oriented language provides Web page authors with the power to reach a very high level of interaction between the user and the document.

What is JavaScript?

JavaScript is an easy-to-use object scripting language designed for creating live online applications that link together objects and resources on both clients and servers. While programmers to create new objects and applets use Java, JavaScript is designed for use by HTML page authors and enterprise application developers to dynamically script the behaviour of objects running on either the client or the server. JavaScript's design and concepts represent the next generation of software for the Internet and is:

Designed for creating network-centric applications

Complementary to and integrated with java

Complementary to and integrated with HTML

Open and cross-platform

With JavaScript, an HTML page might contain a form that processes data on the client side. A server side JavaScript might pull data out of a relational data base and format it in HTML on the fly. A page might contain JavaScript scripts that run on both the client and the server.

JavaScript in a browser:

Client side JavaScript:

Client side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. The browser begins interpreting the HTML code and then encounters a JavaScript script. If the script has no syntax errors, the browsers built in interpreter execute it. The JavaScript client side mechanism features many advantages over traditional CGI server side scripts.

Browser objects:

JavaScript is an object-based programming language. Its built in object model is mostly based on the HTML content of the given Web Page. The tight interaction between JavaScript and other browser objects (such as forms, frames, images, browser windows) provides full control over various page elements and enables the programmer to create a link between "external" objects and "internal" ones. E.g. A JavaScript that invokes a Java applet from an HTML form.

WHAT CAN JAVASCRIPT DO?

JavaScript can be used to create interesting games, banners, etc. but it is mainly used for form validation. It is tightly connected to browser objects, including forms and their elements. Therefore, it provides a great amount of control over forms. A classic form related script is one that validates a form's fields before it is submitted and cancels the submission to the server if an error is found in one of the fields. The advantage is that client side validation is much faster than validation via transmission to the server and back to the client. Client side validation involves loading the script as plain text. Clicking of the Submit button triggers of a JavaScript function to validate the form.

JavaScript for server side applications can be used using Livewire compiler. Server side JavaScript is actually an alternative to traditional CGI programming via Perl and C++.

JAVASCRIPT and JAVA:

JavaScript supports most of Java's expression syntax and basic control flow statements. They are both based on objects, but their implementations differ. In both languages, many built in functions are implemented as properties and methods of various objects.

JavaScript Differs From Java:

JavaScript is interpreted, not compiled. Java is referred to as a compiled language. Since Java is compiled, common user cannot see the actual code behind the program. Nevertheless, when a user comes across a JavaScript script, he can generally see and even copy the code. A compiled program is slower but more efficient than the one that is interpreted directly from a text file. Interpreted languages are convenient to debug and modify by modifying the text file rather than having to recompile it.

Though both are based on objects, their implementations are different. JavaScript supports built in extensible objects, but no classes or inheritance.

Java is strongly typed as opposed to JavaScript, which is loosely typed. That is, when a variable is declared in Java, its data type has to be specified. In JavaScript all variables, are declared in the same way, a variable of one data type can contain a value of different data type elsewhere in the script.

A very important difference between Java and JavaScript is that JavaScript is integrated with and embedded in HTML. Applets (small Java programs) are always distinct from the HTML code and are only invoked by it.

Java is not only a language for creating applets that run on Web Pages, but rather a general purpose language that can create stand-alone applications (without any connection to the Internet or the World Wide Web). Whereas JavaScript is not a stand-alone programming language. It is interpreted & requires a specific application to run it.

In brief, Java and JavaScript are COMPLEMENTARY LANGUAGES.

The Future Of JavaScript:

JavaScript is perceived to advance in two vectors. The first one is in the direction of "server less" CGI scripting. Small databases of information will be embedded in an HTML document and processed by JavaScript scripts. Cutting the server-client communication will sharply reduce the response time, which is still the number one problem in surfing the Internet.

As ActiveX and VRML become more powerful, JavaScript will work in concert with them, resulting in very powerful applications. JavaScript, a cross-platform language, is currently in use alongside many other languages.

3.2.6 SERVLETS:

WHY SERVLETS?

Once you've committed to going hi-tech, the question turns to software. Now, thanks to Jeeves, Java is one of your options. We introduce the Java Server API, also called, in somewhat quaintly British fashion, Jeeves. This API allows programmers to rapidly create connection-oriented server applications using a pre-defined architecture and providing the supporting object classes necessary.

Support service:

The Java Server API is very much designed for the Web, although technically it can be used for any generic TCP/IP services. Jeeves itself is an HTTP or Web server that can be extended to include other protocols and applications. Each Web connection is processed by a servlet. This is similar in concept to a server application, but with minimal overhead and handling only the portions of the service workflow for which it is designed. Programmatically, the servlet is similar to an applet with the exception that there is no need for a graphical interface to the program. It is a piece of Java byte code that can be downloaded from the network if necessary. The Jeeves framework provides all the needs for handling multiple connections, managing a pool of servlets, keeping track of backlog, and dispatching service requests using a given set of classes. Basically, it begins with the main server loading up, binding to a TCP/IP port, and starting a pool of servlets. The server resides on this port and accepts incoming connections, which are placed in a queue for processing by the servlets. Before passing this connection to an awaiting servlet, the server checks the servlet pool to see that enough servlets are available for future requests. The server can maintain a fixed pool of servlets or it can dynamically add and remove servlets from the pool to satisfy the requirements of the application. The server will pass a request only to a servlet of the appropriate type for handling the request. There can be several pools of servlets, each a different type of request handler. For example, one servlet can handle HTML file requests, another can serve CGI requests, and still others can handle application-specific requests. Each servlet is designed to handle only its own area of service and thus does not need the huge overhead of traditional Web servers, where each server process needs to know how to handle everything. This makes for compact programs that can execute in a fast and efficient manner, supported by other programs that can handle other tasks more suited to them.

SERVLTS AGAIN! - More about them:

According to JavaSoft:
Servlets are modules that run inside request/response-oriented servers, such as Java-enabled web servers, and extend them in some manner. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database. Servlets are to servers what applets are to browsers.

Example Uses:

A few of the many applications for servlets include,
Processing data POSTed over HTTPS using an HTML form, including purchase order or credit card data. A servlet like this could be part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

Allowing collaboration between people. A servlet can handle multiple requests concurrently; they can synchronize requests to support systems such as on-line conferencing.

Forwarding requests. Servlets can forward requests to other services and servlets. This allows them to be used to balance load among several servers that mirror the same content. It also allows them to be used to partition a single logical service over several servers, according to task type or organizational boundaries.

Being a community of active agents. A servlet writer could define active agents that share work among each other. Each agent would be a servlet, and the agents could pass data among themselves.

The Servlet API, which we use to write servlets, assumes nothing about how a servlet is loaded, the server environment in which the servlet runs, or the protocol used to transmit data to and from the user. This allows servlets to be embedded in many different web servers.

Servlets are an effective substitute for CGI scripts: they provide a way to generate dynamic documents that is both easier to write and faster to run. They also address the problem of doing server-side programming with platform-specific APIs. Servlets are developed with the Java Servlet API, a standard Java extension. While it is not part of the core Java framework, which must always be part of all products bearing the Java brand, it will be made available with such products by their vendors as an add-on package. Many popular web servers already support it.

In some ways, a servlet is similar to an applet. An applet is a chunk of Java code that executes under control of a browser. A servlet is a chunk of Java code that executes under control of a server program. You must run your servlet under the control of a Java-enabled server program like servletrunner.

Server-Side JavaScript:

Server-Side JavaScript is another solution for implementing dynamic web sites. It lets you embed JavaScript into precompiled HTML pages. By pre-compiling the web pages you improve performance, but the only servers that implement Server-Side JavaScript is Netscape's Enterprise and FastTrack Servers. This again ties you to a particular vendor.

JAVA SERVLETS – Where They score:

Java servlets are one of the most exciting new technologies I have had the opportunity to work with. Servlets are efficient, persistent, portable, robust, extensible, secure, and receiving widespread acceptance. If you only use them to replace CGI, you will have saved yourself a lot of time and headache. Servlets solve many of the common problems you run into when using CGI. And they prove to have a clear advantage over many of the other alternatives.

Efficient:

A servlet's initialization code is only executed the first time the web server loads it. Once the servlet is loaded, it is only a matter of calling a service method to handle new requests. This is a much more efficient technique than loading a completely new executable with every request.

Persistent:

Servlets can maintain state between requests. Once a servlet is loaded it stays resident in memory while serving incoming requests. A simple example of this would be a Vector that holds a list of categories used in an online catalog. When the servlet is initialized it queries the database for a list of categories and stores these categories in a vector. As it services requests the servlet accesses the Vector that holds the categories instead of querying the database again. Taking advantage of the persistent characteristics of servlets can improve your applications performance drastically.

Portable:

Servlets are developed using Java therefore they are portable. This gives servlets the ability to be moved to a new operating system without changing the source. You can take code that was compiled on a Windows NT platform and move it to a Solaris box without making any changes.

Robust:

Because servlets are developed with access to the entire JDK, they are a very powerful and robust solution. Java provides you with a very well defined exception hierarchy for error handling. It has a garbage collector to prevent problems with memory leaks. In addition it includes a very large class library that includes network support, file support, database access, distributed object components, security, and many other classes.

Extensible:

Another advantage servlets gain by being developed in an object- oriented language like Java is they can be extended and polymorphed into new objects that better suit your needs. A good example of this is an online catalog. You may want to display the same catalog search tool at the top of every dynamic page throughout your web site. You definitely don't want to add this code to every one of your servlets. So you implement a base servlet that builds and initializes the search tool and then extend it to display transaction specific responses.

Secure:

Servlets run on the server side inheriting the security provided by the web server. Servlets can also take advantage of the Java Security Manager.

Widespread Acceptance:

Because of all there is to be gained from using Java servlets, they are being widely accepted. Vendors are providing servlet support in two main forms. The first form being servers that have built in support for servlets, and the second is by using third-party add-ons. The following tables list the vendors and their associated products.

Servers with built in Servlet Support:

Product	Vendor
Acme.Serve	Acme Java Software
Apache Web Server	Apache
Domino Go Web Server	Lotus
Dynamo Application Server	ATG
Enterprise Server	Netscape
Enterprise Server	KonaSoft
Internet Connection Server	IBM
iTP Web Server	Tandem
Java Web Server	Sun Microsystems
Jetty	Mort Bay
Jigsaw Server	World Wide Web Consortium
NetForge	Novocode
ServletFactory	Early Morning Software
Sun Web Server	Sun Microsystems
Tengah Application Server	WebLogic
Visual Age WebRunner Toolkit	IBM
WEASAL	Web Easy
WebCore	Paralogic
Website Professional	O'Reilly
Zeus Web Server	Zeus Technology

Third-Party Add-Ons with Servlet Support:

Product	Vendor
Servletrunner	Sun Micro Systems
Jrun	Live Software
ServletExec	New Atlanta
Servlet CGI Development Kit	Unicom
WebSphere Application Server	IBM
WAICoolRunner	Gefion Software

RUNNING SERVLETS – HOW?

The **servletrunner** is a small utility, intended for testing. It is multithreaded, so it can run more than one servlet. It can be used therefore, to run multiple servlets simultaneously, or to test one servlet that calls other servlets in order to satisfy client requests. Unlike some web servers, it does not automatically reload servlets when they are updated. What this means is that once you run a servlet in **servletrunner**, simply replacing the servlet class file with a new version of the class file does not cause **servletrunner** to forget the original version. It will continue to execute the original version even if you provide a new class file. To cause it to load the new version, you must terminate and restart **servletrunner**. Because it is small, however, there is very little overhead associated with stopping and restarting it in order to use a new version of a servlet.

You can obtain a usage message about servletrunner by starting it with the -help flag at the command prompt.

```
Servletrunner -help
Usage: servletrunner [options]
Options:
  -p port      the port number to listen on
  -b backlog  the listen backlog
  -m max       maximum number of connection handlers
  -t timeout  connection timeout in milliseconds
  -d dir       servlet directory
  -s filename servlet property file name
```

You can control the values for any of these parameters by entering command-line arguments when you start the program. There are default values for each parameter. The default values are displayed if you simply start the program running with no command-line arguments

```
Servletrunner starting with settings:
  port = 8080
  backlog = 50
  max handlers = 100
  timeout = 5000
  servlet dir = .\examples
  document dir = .\examples
  servlet propfile = .\examples\servlet.properties
```

servlet dir :

This is the name and location of the folder that must contain the class file for your servlet. You can copy your class files to that folder, or you can change that setting to make it match the directory that contains your class file when you start the program running.

There is more than one way to execute a servlet. One way is to call the servlet directly by putting its URL in your browser. The URL for a servlet has the following general form: *http://machine-name:port/servlet/servlet-name*

An alternative URL used in the browser to execute a sample servlet is

http://localhost:8080/servlet/Servlet01

where:

- **localhost** indicates that servletrunner is running in a separate process on the same machine
- The name of the servlet class file is **Servlet01.class**

In either case, the servletrunner program must be running before the servlet is executed. In the URL, '**/servlet/**' has a special meaning. Unlike other URLs, this is not the actual name of a directory on the server machine. Rather, it tells the server program to invoke the servlet whose name follows. Another way of invoking a servlet is by calling it by using the 'Submit' button in a HTML form.

Servlet Architecture Overview:

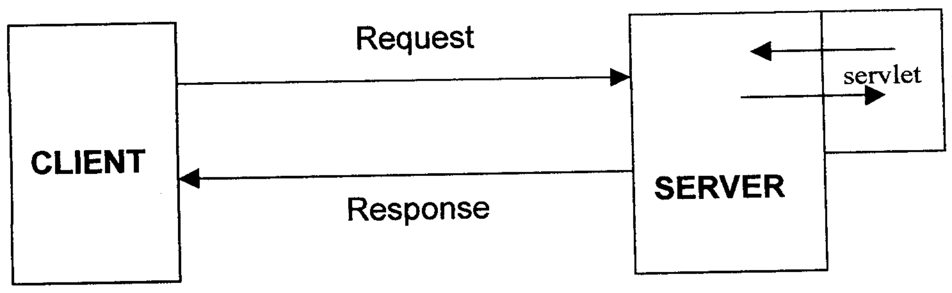
The central abstraction in the JSDK is the Servlet interface. All servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet. The Servlet interface provides for methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

When a servlet accepts a call from a client it receives two objects: one is a ServletRequest and the other is a ServletResponse. The ServletRequest class encapsulates the communication from the client to the server, while the ServletResponse class encapsulates the communication from the servlet back to the client.

The ServletRequest interface allows the servlet access to information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it. It also provides the servlet with access to the input stream, ServletInputStream, through which the servlet gets data from clients that are using application protocols such as the HTTP POST and PUT methods. Subclasses of ServletRequest allow the servlet to retrieve more protocol-specific data. For example, HttpServletRequest contains methods for accessing HTTP-specific header information.

The ServletResponse interface gives the servlet methods for replying to the client. It allows the servlet to set the content length and mime type of the reply, and provides an output stream, ServletOutputStream, and a Writer through which the servlet can send the reply data. Subclasses of ServletResponse give the servlet more protocol-specific capabilities. For example, HttpServletResponse contains methods that allow the servlet to manipulate HTTP-specific header information.

HTTP servlets have some additional objects that provide session-tracking capabilities. The servlet writer can use these APIs to maintain state between the servlet and the client that persists across multiple connections during some time period.



Servlet Lifecycle:

The life cycle of a Java Servlet is a very simple object-oriented design. A servlet is constructed and initialized. It then services zero or more requests until the service that it extends shuts down. At this point the servlet is destroyed and garbage collected. This design explains why servlets are such a good replacement for CGI. The servlet is loaded only once and it stays resident in memory while servicing requests.

The interface that declares this framework is the javax.servlet.Servlet interface. The Servlet interface defines the life cycle methods. These methods are the init(), the service(), and the destroy() methods.

init()

The init() method is where the servlet's life begins. The server calls it immediately after the servlet is instantiated. It is called only once. In the init() method the servlet creates and initializes the resources that it will be using while handling requests. The init() method's signature is defined as follows.

```
public void init(ServletConfig config) throws ServletException;
```

The init() method takes a ServletConfig object as a parameter. You should save this object, so that it can be referenced later. The most common way of doing this is to have the init() method call super.init() passing it the ServletConfig object. You will also notice that the init() method can throw a ServletException. If, for some reason, the servlet cannot initialize the resources necessary to handle requests, the init() method should throw a ServletException.

service()

The service() method handles all request sent by a client. It cannot start servicing requests until the init() method has been executed. You will not usually implement this method directly, unless you extend the GenericServlet abstract class. The most common

implementation of the service() method is in the HttpServlet class. The HttpServlet class implements the Servlet interface by extending GenericServlet. Its service() method supports standard HTTP/1.1 requests by determining the request type and calling the appropriate method. The signature of the service() method is shown below.

```
public void service(ServletRequest req, ServletResponse res)
throws ServletException, IOException;
```

The service() method implements a request and response paradigm. The ServletRequest object contains information about the service request, encapsulating information provided by the client. The ServletResponse object contains the information returned to the client.

destroy()

This is the method that signifies the end of a servlet's life. When a service is being shutdown it calls the servlet's destroy() method. This is where any resources that were created in the init() method should be cleaned up. If you have an open database connection, you should close it here. This is also a good place to save any persistent information that will be used the next time the servlet is loaded. The signature of the destroy() is very simple, but it is displayed below just to complete the picture.

```
public void destroy();
```

Intracting with clients:

Servlet writers who are developing HTTP servlets that specialize the HttpServlet class should override the method or methods designed to handle the HTTP interactions that their servlet will handle. The candidate methods include,

- doGet, for handling GET, conditional GET and HEAD requests
- doPost, for handling POST requests
- doPut, for handling PUT requests
- doDelete, for handling DELETE requests

A HttpServletRequest object provides access to HTTP header data, such as any cookies found in the request and the HTTP method with which the request was made. It, of course, allows you to obtain the arguments that the client sent as part of the request. How you access the client data might depend on the HTTP method of the request.

- For any HTTP method, you can use the getParameterValues method, which will return the value of a named parameter. (The method getParameterNames provides the names of the parameters.) You can also manually parse the request.
- For requests using the HTTP GET method, the getQueryString method will return a String to be parsed.
- For HTTP methods POST, PUT, and DELETE, you have the choice between two methods. If you expect text data, then it can be read using the BufferedReader returned by the getReader method; if you expect binary data, then it should be read with the ServletInputStream returned by the getInputStream method.

Note that you should use either the getParameterValues method or one of the methods that allow you to parse the data yourself. They cannot be used together in a single request. For responding to the client, an HttpServletResponse object provides two ways of returning the response data to the user. You can use the writer returned by the getWriter method or the output stream returned by the getOutputStream method. You should use getWriter to return text data to the user, and getOutputStream for binary data. Before accessing the writer or output stream, HTTP header data should be set.

3.2.7 JDBC:

JDBC is a Java API for executing SQL statements. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. In other words, with the JDBC API, it isn't necessary to write one program to access a Sybase database, another program to access an Oracle database, another program to access an Informix database, and so on. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. And, with an application written in the Java programming language, one also doesn't have to worry about writing different applications to run on different platforms. The combination of Java and JDBC lets a programmer write it once and run it anywhere. (We explain more about this later.)

Java, being robust, secure, easy to use, easy to understand, and automatically downloadable on a network, is an excellent language basis for database applications. What is needed is a way for Java applications to talk to a variety of different databases. JDBC is the mechanism for doing this.

JDBC extends what can be done in Java. For example, with Java and the JDBC API, it is possible to publish a web page containing an applet that uses information obtained from a remote database. Or an enterprise can use JDBC to connect all its employees (even if they are using a conglomeration of Windows, Macintosh, and UNIX machines) to one or more internal databases via an Intranet. With more and more programmers using the Java programming language, the need for easy database access from Java is continuing to grow.

MIS managers like the combination of Java and JDBC because it makes disseminating information easy and economical. Businesses can continue to use their installed databases and access information easily even if it is stored on different database management systems. Development time for new applications is short. Installation and version control are greatly simplified. A programmer can write an application or an update once, put it on the server, and everybody has access to the latest version. And for businesses selling information services, Java and JDBC offer a better way of getting out information updates to external customers.

What does JDBC do?

Simply put, JDBC makes it possible to do three things:

Establish a connection with a database

Send SQL statements

Process the results.

The following code fragment gives a basic example of these three steps:

```
Connection con = DriverManager.getConnection (
          "jdbc:odbc:wombat", "login", "password");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");
while (rs.next()) {
   int x = getInt("a");
   String s = getString("b");
   float f = getFloat("c");
}
```

JDBC is a Low-level API and a base for High-level API's:

JDBC is a ``low-level" interface, which means that it is used to invoke (or "call") SQL commands directly. It works very well in this capacity and is easier to use than other database connectivity APIs, but it was designed also to be a base upon which to build higher-level interfaces and tools. A higher-level interface is ``user-friendly," using a more understandable or more convenient API that is translated behind the scenes into a low-level interface such as JDBC. At the time of this writing, two kinds of higher-level APIs are under development on top of JDBC:

An embedded SQL for Java. At least one vendor plans to build this. DBMSs implement SQL, a language designed specifically for use with databases. JDBC requires that the SQL statements be passed as strings to Java methods. An embedded SQL preprocessor allows a programmer to instead mix SQL statements directly with Java: for example, a Java variable can be used in an SQL statement to receive or provide SQL values. The embedded SQL preprocessor then translates this Java/SQL mix into Java with JDBC calls.

a direct mapping of relational database tables to Java classes. JavaSoft and others have announced plans to implement this. In this ``object/relational" mapping, each row of the table becomes an instance of that class, and each column value corresponds to an attribute of that instance. Programmers can then operate directly on Java objects; the required SQL calls to fetch and store data are automatically generated ``beneath the covers". More sophisticated mappings are also provided, for example, where rows of multiple tables are combined in a Java class.

As interest in JDBC has grown, more developers have been working on JDBC-based tools to make building programs easier, as well. Programmers have also been writing applications that make accessing a database easier for the end user. For example, an application might present a menu of database tasks from which to choose. After a task is selected, the application presents prompts and blanks for filling in information needed to carry out the selected task. With the requested input typed in, the application then automatically invokes the necessary SQL commands. With the help of such an application, users can perform database tasks even when they have little or no knowledge of SQL syntax.

JDBC Vs ODBC:

At this point, Microsoft's ODBC (Open database Connectivity) API is probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms. So why not just use ODBC from Java?

The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes, ``Why do you need JDBC?" There are several answers to this question:

ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.

A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer ``void *". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.

ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.

A Java API like JDBC is needed in order to enable a ``pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

In summary, the JDBC API is a natural Java interface to the basic SQL abstractions and concepts. It builds on ODBC rather than starting from scratch, so programmers familiar with ODBC will find it very easy to learn JDBC. JDBC retains the basic design features of ODBC; in fact, both interfaces are based on the X/Open SQL CLI (Call Level

Interface). The big difference is that JDBC builds on and reinforces the style and virtues of Java, and, of course, it is easy to use.

More recently, Microsoft has introduced new APIs beyond ODBC: RDO, ADO, DAO, and OLE DB. These designs move in the same direction as JDBC in many ways, for example, in being object-oriented interfaces to databases based on classes that can be implemented on ODBC. However, we did not see functionality in any of these interfaces compelling enough to make them an alternative basis to ODBC, especially with the ODBC driver market well-established. Mostly they represent a thin veneer on ODBC. This is not to say that JDBC does not need to evolve from the initial release described in this book; we discuss a few directions in ``The Future of JDBC" in Appendix B. However, we feel that most new functionality belongs in higher-level APIs such as the object/relational mappings and embedded SQL we mentioned earlier in ``JDBC Is a Low-level API and a Base for Higher-level APIs."

Javasoft Framework:

JavaSoft provides three JDBC product components:

The JDBC driver manager (included as part of the Java Development Kit (JDK))

The JDBC driver test suite (available from the JDBC web site)

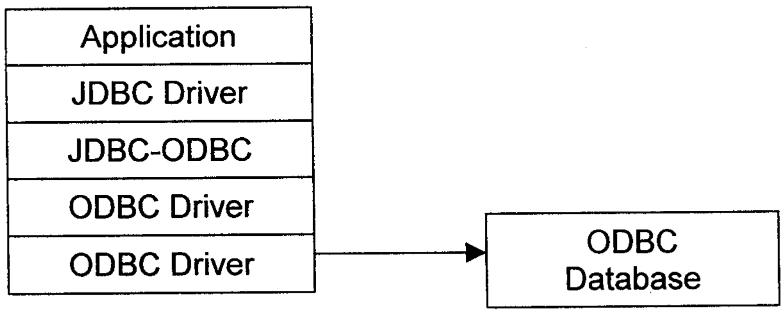
The JDBC-ODBC bridge (included in the Solaris and Windows versions of the JDK)

The JDBC driver manager is the backbone of the JDBC architecture. It actually is quite small and simple; its primary function is to connect Java applications to the correct JDBC driver and then get out of the way.

The JDBC driver test suite provides some confidence that JDBC drivers will run your program. Only drivers that pass the JDBC driver test suite can be designated JDBC Compliant(TM).

The JDBC-ODBC bridge allows ODBC drivers to be used as JDBC drivers. It was implemented as a way to get JDBC off the ground quickly, and long term will provide a way to access some of the less popular DBMSs if JDBC drivers are not implemented for them.

JDBC – ODBC Bridge Driver



JDBC Driver Types:

The JDBC drivers that we are aware of at this time generally fit into one of four categories:

JDBC-ODBC bridge plus ODBC driver:

The JavaSoft bridge product provides JDBC access via ODBC drivers. Note that ODBC binary code, and in many cases database client code, must be loaded on each client machine that uses this driver. As a result, this kind of driver is most appropriate on a corporate network where client installations are not a major problem, or for application server code written in Java in a three-tier architecture.

Native-API partly-Java driver: This kind of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

JDBC-Net pure Java driver: This driver translates JDBC calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect its pure Java clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC alternative. It is likely that all vendors of this solution will provide products suitable for Intranet use. In order for these products to also support Internet access, they must handle the additional requirements for security, access through firewalls, and so forth, that the Web imposes.

Native-protocol pure Java driver: This kind of driver converts JDBC calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is an excellent solution for Intranet access. Since many of these protocols are proprietary, the database vendors themselves will be the primary source. Several database vendors have these in progress.

The expectation is that eventually driver categories 3 and 4 will be the preferred way to access databases from JDBC. Driver categories 1 and 2 are interim solutions where direct pure Java drivers are not yet available. There are possible variations on categories 1 and 2 (not shown in the table below) that require a connector, but these are generally less desirable solutions. Categories 3 and 4 offer all the advantages of Java, including automatic installation (for example, downloading the JDBC driver with an applet that uses it)

SYSTEM DESIGN & DEVELOPMENT

4.System Design & Development:

Introduction:

System design is a solution to " how to " approach to the creation of a new system. This important phase is composed of several steps. It provides the understanding and procedural details necessary for implementing the system. Emphasis must be laid on translating the performance requirements into design specifications. Design goes through logical and physical stages of development. Logical design reviews the present physical system, prepares input and output specifications, and makes edit, security and control specifications. The physical system and plans the system implementation. Design specification instruct programmers about the system should do. The programmers in turn write programs that accept from us, process data, procedure the reports, and store data in the files.

SUPPORT BUSINESS ACTIVITIES:

A fundamental objective in the design of an interactive system is to ensure that it supports the activity for which it is developed. For example, if it is essential for an organization to move information very quickly to remain competitive, then the design specification of the system must be based around this essential business objective.

Elements of system design:

The components of Question Bank system described during requirement analysis are the focal point in system design. Analysts must design the following elements:

Data flows

The movement of data into, around, and out of the system

Data stores

Temporary or permanent collections of data

Process

Activities to accept, manipulate, and deliver data and information

Procedures

Methods and routines for using the QB system to achieve the intended results

Controls:

Standards and guidelines for determining whether activities are occurring in the anticipated or accepted manner, that is, "under control"

Roles:

The responsibilities of all persons involved with the new system, including end-users, computer operators, and support personnel.

4.1 I/P Design:

A major step in the design in the proper preparation of input. Inaccurate input data are the most common cause of errors in data processing. Input design is the process of converting the user generated inputs into computer based format. The objective of designing input data format is to make data entry task as easy free from errors as possible. The allotted space for each field, the field sequence and the format in which the data fields are entered (e.g., dd/mm/yy) for date fields were decided.

Design of user interface:

The primary interface of the system with the user is through internet explorer(links etc..). Through the url links the user can surf the Examination part.

4.2 O/P Design:

Computer output is the most important and the direct source of information to the user. Efficient, intelligent output design should improve the system's relationship with the user and help in decision making. During the design of the output devices are identified. For the system the major form of the output is the printer.

4.3 Database Design:

By studying the existing system and conducting preliminary data flow diagrams, the database developed using normalization ensures that the database is well defined and to store the necessary data.

DATABASE NAME: TBLEXAMINEE

NAME	TYPE	SIZE
ExamineeID	Number	10
ExamineeName	Text	30
Password	Text	5
EmailID	Text	50
Qualification	Text	20
Subject	Text	8
Age	Number	3
Contact Address	Text	120

DATABASE NAME : TBLSUBJECT

NAME	TYPE	SIZE
SubjectID	Varchar2	4
Subject	Text	50

DATABASE NAME : TBLQUESTION

NAME	TYPE	SIZE
QuestionID	Number(Long)	4
SubjectID	Number(Long)	4
QuestionText	Text	255
Keyword	Text	50

DATABASE NAME : TBLANSWER

NAME	TYPE	SIZE
QuestionID	Number (Long)	4
ChoiceNumber	Number (Byte)	1
Choice	Text	255
IsCorrect	Yes/No	1

DATABASE NAME : TBLEXAMINATIONSCHEDULE

NAME	TYPE	SIZE
ScheduleID	Number (Long)	4
QuestionPaperID	Number (Long)	4
ExamineeID	Number (Long)	4
StartDate	Date/Time	8
LastDate	Date/Time	8
HasAppeared	Yes/No	1
AppearedDate	Date/Time	8
AppearedTime	Date/Time	8
FinishTime	Date/Time	8
PercentageObtained	Number(Single)	4
Result	Text	1

DATABASE NAME : TBLEXAMINEEANSWERS

NAME	TYPE	SIZE
ScheduleID	Number (Long)	4
QuestionID	Number (Long)	4
ExamineeID	Number (Long)	4
ChoiceNumber	Number (Byte)	1
ExamineeAnswer	Yes/No	1
IsCorrect	Yes/No	1

DATABASE NAME : TBLEXAMINEESCORE

NAME	TYPE	SIZE
ScheduleID	Number (Long)	4
QuestionID	Number (Long)	4
ExamineeID	Number (Long)	4
Securedscore	Number	1

4.4 Process Design:

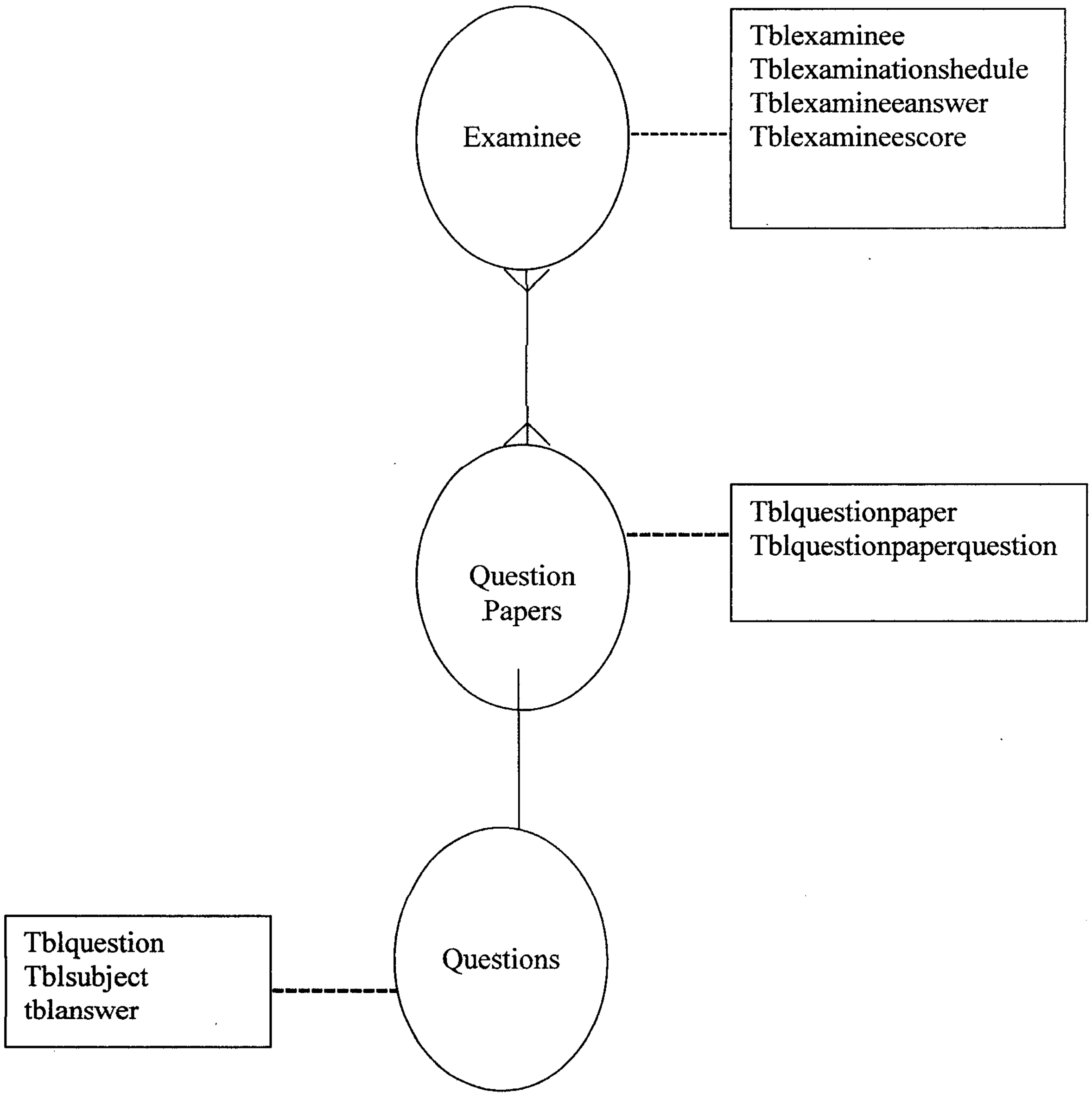
Data Flow Analysis:

The data flow diagram is pictorial representation of the working of the system. This takes very important role in the system analysis part to know how the existing system is going on what modification to be done to overcome the problem occurs in the system.

A data flow diagram (DFD) is also known as "bubble chart", has the purpose of clarifying system requirements and identifying major transformation that will become program in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail.

A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flow in the system. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

DATA FLOW DIAGRAM



SYSTEM IMPLEMENTATION & TESTING

5.1System Implementation:

System implementation is the process of making the newly designed system fully operational. The system is implemented after careful testing.

The following steps have been followed in the implementation of this system.

Implementation planning

Equipment acquisition and installation.

System conversion

User Training

Implementation Planning:

A logical starting point for this type of planning involves knowledge of the following areas.

- ❖ Personal needs.
- ❖ Programming equipment's selected.
- ❖ Physical requirements
- ❖ Conversion activities.

Initially a primary implementation plan is prepared to schedule and manage many different activities that must be completed for a successful system implementation. The preliminary plan serves as a basis for checking the availability of resources for implementation activities. A complete implementation plan includes following items.

- System training plan.
- System test plan.
- System conversion plan.
- Overall implementation plan.

System conversion:

The system implementation could be done in the following methods.

- ♦ Direct conversion method.
- ♦ Parallel conversion method.
- ♦ Phase-in method.

The system is implemented by using parallel conversion and phase-in method. Parallel conversion is a method in which both the old and new system is run parallel for a particular period. In the phase-in method, the new system is implemented module by module.

User Training:

The training should include every one associated with the implementation, use, operation or maintenance of new system. Hands-on training to the lab staff is essential to make them comfortable with the system. Accordingly, classroom lectures about the system have been given to the lab staff.

5.2System Testing:

As we have explained before, Software Development consists of four phases i.e., Requirements Analysis and Specifications, Design, Coding, Testing and Implementation. We can define Testing as a very critical role for quality assurance and for ensuring the reliability of software. Though there are several kinds of tests, we adopted a few.

Types Of Testing Done:

Unit Testing:

Here the first level of testing is unit testing. In this, we are going to test different modules against the specifications produced in during design. Unit testing is essentially for verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules.

Integration Testing:

The goal here is to see if the modules can be integrated properly, The emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design, and hence the emphasis is on testing module interactions.

System Testing:

Here the goal is to test whether the software meets its overall requirements. We have taken care all points with respect to Requirement Specification Document, which was already presented before.

Functional Testing:

There are two basic approaches to testing, functional and structural. In functional testing the structure of the program is not considered. This functional testing is also referred as **black box testing.** In the structural approach, on the actual code of the program or to module to be tested. This Structural approach is also called as glass box testing(White box testing)

During system testing, the system is used experimentally to ensure that the software does not fail, i.e. that it will run according to its specifications and in the way the uses expect. Special test data are input for processing, and the results are examined.

In many organizations, persons other than those who wrote the original programs to ensure more complete and unbiased testing and more reliable software perform testing.

The programs are tested with sample data supplied by the user and necessary corrections to the programs were carried out if any errors are found. All the reports are to be checked and approved by the user. The system is very user-friendly with display messages.

The operational manuals supplied along with the computer system by the manufacturers are to be referred to by the user as and when he needs it.

The steps involved in testing and debugging of system are

Live data's were inputted to system

Testing the modules independently to check their efficiency and correctness

If any error occur program coding is checked and rectified

Making necessary changes to the system as desired by the user.

After achieving the desired results during testing the system, the process goes for implementation.

REFINEMENTS BASED ON FEEDBACK

Refinements based on feedback:

The system life cycle does not finish all on a sudden after implementation. The HR who works on the system regularly finds the problems later. The problems and possible suggestions are conveyed to the system development department and the department together decides corrective measures. These corrections are incorporated to the system to make it more efficient and user friendly.

This process repeats itself for some more time. The system developers have to be touch with the system even after implementation and changeover because of these refinements based on the feedback from the user.

CONCLUSION

7.Conclusion

Since the earlier system is done manually the assistant of the company finds it very difficult to update and keep track of all the details. As a result of that it takes more time. As the system is computerized the user feels that the data are given to the system it immediately gives the result.

This system makes the job easier and reduces the workload of the HR Manager. Once the test has been taken by the examinee, since his results were evaluated by the system itself.

SCOPE FOR FUTURE DEVELOPMENT

8.Scope for future development

As per the requirements of the HR department the system can be updated. Since Java supports Graphical user interface it can be made a point that future enhancements can be added based on the requirements of the HR persons.

The candidate's facility is spotted first and changes are made so that recruitment can be done easily at any place at any time. The inconvenience felt by the candidates will immediately attended.

BIBLIOGRAPHY

Bibliography:

- Patric Naughton and Herbert Schild-Java 2
- The complete Reference-3rd edition
- E.Balagurusamy-Programming with java
- George Reese Database programming with JDBC and Java
- www.Brainbench.com
- www.sun.java.com
- Java-Hcl infosystems Ltd
- HTML-Vogue software services
- JavaScript- Vogue software services

APPENDIX-A

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;

import java.sql.*;


public class disp1 extends HttpServlet

{

        Connection con;

        Statement st;

        ResultSet rs;

        String sr,s,s2,s3,s4,s5,s6,s1,s9,s10,to,ra,i,s7;

        float str;

        double b,a;

        long ti=0,t=60000,la,ct,a1=0,a2=0,a3,a4=0;

        int c=0,s8,s11,j,z=1,b1,z1;

     int v[]=new int[100];


        public void init(ServletConfig cfg)throws ServletException

        {

        try

         {

                super.init(cfg);
```

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;

import java.sql.*;


public class disp1 extends HttpServlet

{

        Connection con;

        Statement st;

        ResultSet rs;

        String sr,s,s2,s3,s4,s5,s6,s1,s9,s10,to,ra,i,s7;

        float str;

        double b,a;

        long ti=0,t=60000,la,ct,a1=0,a2=0,a3,a4=0;

        int c=0,s8,s11,j,z=1,b1,z1;

      int v[]=new int[100];


        public void init(ServletConfig cfg)throws ServletException

        {

        try

         {

                super.init(cfg);
                }catch(Exception e) {}
        }


        public void doGet(HttpServletRequest req,HttpServletResponse res) throws
ServletException,IOException
        {
        try
                {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con=DriverManager.getConnection("jdbc:odbc:test");
st=con.createStatement();
                res.setContentType("text/html");
                PrintWriter out=res.getWriter();
String module=req.getParameter("module");
String b2=req.getParameter("b1");


b=Integer.parseInt(b2);


String z2=req.getParameter("z1");


z=Integer.parseInt(z2);


String s20=module.substring(0,1);
```

```
/*String i=req.getParameter("i");
j=Integer.parseInt(i);*/
  HttpSession session=req.getSession(true);
 Integer hit=(Integer)session.getValue("Hit Counts");
a3=t/1000;
        if(hit==null)
        {
                        ct=session.getCreationTime();
                        //System.out.println("time start :"+ct);

        out.println("<h2><center><u>"+module+"FAQs</u></center></h2>");
                        //out.println("<b><marquee behaviour=slide>Time Alloted is
 <u>15 </u>minutes</marquee></b>");
                        hit=new Integer(1);
        }
                else

                        hit=new Integer(hit.intValue()+1);
                        session.putValue("HitCounts",hit);
                        la=session.getLastAccessedTime();
                        System.out.println("Last time " +la);
                        ti=la-ct;
                         //  ti=ti+la;
```

```
          //System.out.println(ti);
     /*if(ti>t)
          {
                    out.println("<h2>Time Slot  over</h2>");
                    out.println("<br>Your score is :<h3>"+c+"</h3>");
                    if (c>=2)
                    {
                              out.println("congrats");
                    }
                    else
                    {
                              out.println("better luck next time");
                    }
          }
          else
          {*/
                    out.println("<html><body bgcolor=lightblue text=black>");
                    a1=ti/1000;
                    a2=a3-a1;
        a4=a4+a1;
                    //out.println("Time Starts Now "+(ti/1000));
                    out.println("<br>Still  "+a2+" secs more");
                    if((hit.intValue())==11)
        {
                out.println("Result is <br><h3>"+c+"</h3>");
                if(c>=2)
                {
                        out.println("Congratz");
                        out.println("Passed thru");
                }
                else
                {
                        out.println("better luck next time");
                }
        }
        else
        {
//out.println("test");
                //a=Math.random();
                //b=Math.round(30*a);
                //out.println(b);
                b1=(int)b;
                        v[z]=(int)b;
  /* for(int i=0;i<=z;i++)
                {
                        out.println(v[z]);
                }*/


                        for(int i=0;i<=z;i++)

                        {

                        for(int j=i+1;j<=z;j++)

                        {

                            if(v[i]==v[j])

                            {

                                b=30-i;

                            }

                        }

                }


                        s=s20+z;

                        //System.out.println(s);

out.println("<form name=f1 method=get action='http://localhost:8080/servlet/disp1'>");

                        rs=st.executeQuery("Select * from question where

questionno='"+s+"'");

while(rs.next())

                        {

                            s7=rs.getString(2);

                            s1=rs.getString(4);

                            s2=rs.getString(5);
```

```
                    out.println("better luck next time");
               }



                              }
          if(z<21)
                    {



out.println("<h3>"+z+")"+"&nbsp;"+s1+"</h3><br><h4>");
out.println("<input type=radio name=r value=a> "+s2+"<br>");
out.println("<input type=radio name=r value=b> "+s3+"<br>");
out.println("<input type=radio name=r value=c> "+s4+"<br>");
out.println("<input type=radio name=r value=d> "+s5+"<br>");
out.println("<br><br><input type=submit value=submit></h4>");
out.println("<input type=hidden name=module value="+module+"><br>");
out.println("<br><br><input type=hidden name=b1 value="+b1+">");
out.println("<br><br><input type=hidden name=z1 value="+z1+"></form>");
}
  //out.println(b1);
     //out.println(z1);
j=j+1;
                         if((req.getParameter("r"))!=null)
```

```
                                        {
                                                  s9=req.getParameter("r");
                                                  if(s9.equals(s6))
                                                  c=c+2;
                                        }
                                        else
                                        c=c;
                              //}
                    }

          out.println("</body></html>");
          out.close();
          rs.close();
          st.close();
          }catch(Exception e)
          {
                    System.out.println("Exception is:"+e);
          }
}
public void destroy()
{
          try
          {
```

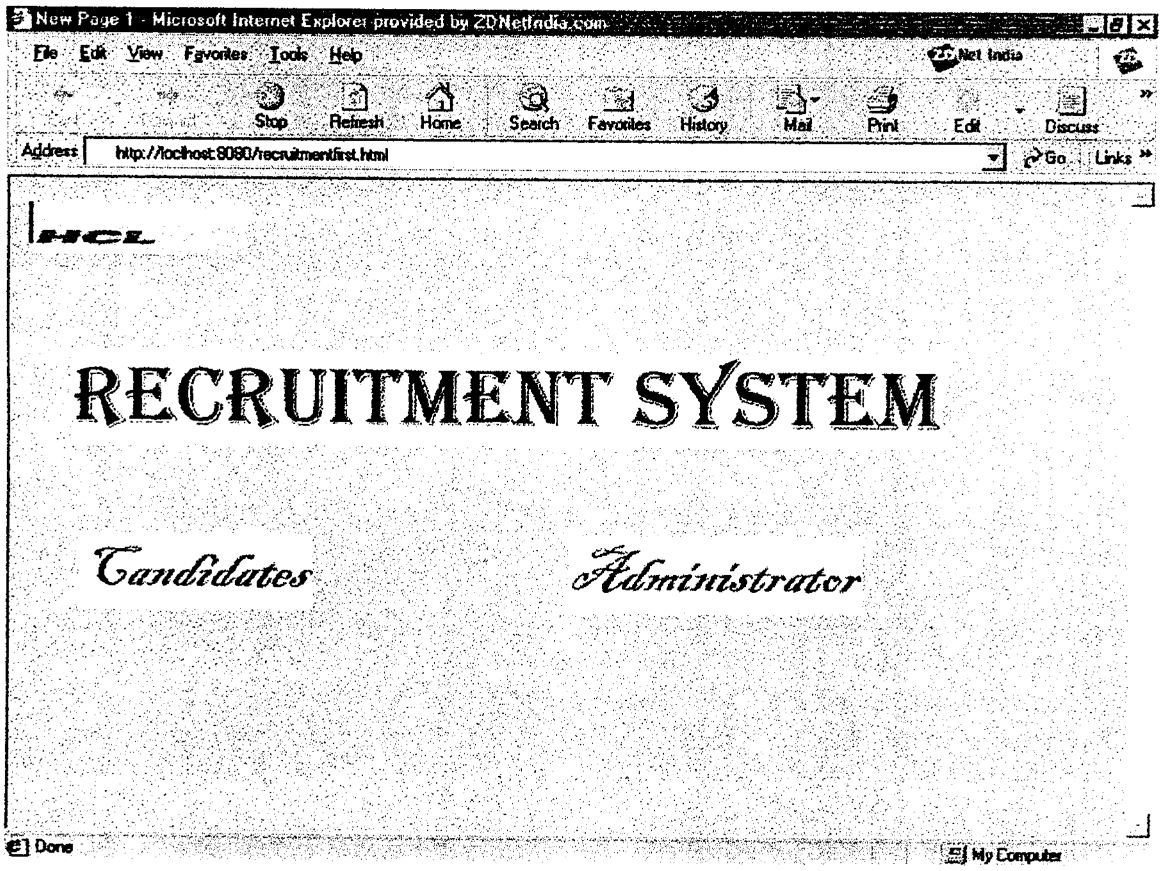
```
                    while(con!=null)
                    con.close();
          }catch(Exception e)
          {
                    System.out.println(e);
          }
     }
}


// Program To Connect JDBC and Servlets


import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;


public class login1 extends HttpServlet
 {
  public void doPost(HttpServletRequest req,HttpServletResponse res) throws
ServletException,IOException
   {
    Connection con=null;
    Statement smt=null;
```

```
PrintWriter out=res.getWriter();


String s1=req.getParameter("id");
String s2=req.getParameter("pas");
if((s1.equals("Sam"))  && (s2.equals("admindiv")))
{
                String r="c:/subha/admindiv.html";
             res.sendRedirect(r);
             System.out.println("after admin");
}


int b=0;
try
 {
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con=DriverManager.getConnection("jdbc:odbc:test");
           smt=con.createStatement();
           ResultSet rs=smt.executeQuery("Select * from register");
           out.println("<html>");
           System.out.println("into rs");
           while(rs.next())
           {
                System.out.println("into while");
```

```
          if((s1.equals(rs.getString(12))) && (s2.equals(rs.getString(13))))
          {
           b=1; System.out.println("into break");break;
             }
}

if(b==1)
{
      String r="c:/subha/aftreg.html";
      res.sendRedirect(r);
      System.out.println("case true");
   }
 else
   {
    System.out.println("case false");
   String t="c:/subha/error.html";
   res.sendRedirect(t);
   }
System.out.println("before catch");
}catch(Exception E)
{
    System.out.println(E);
     E.printStackTrace();
```

```
        }
    out.println("</html>");
    out.close();
  }
}
```

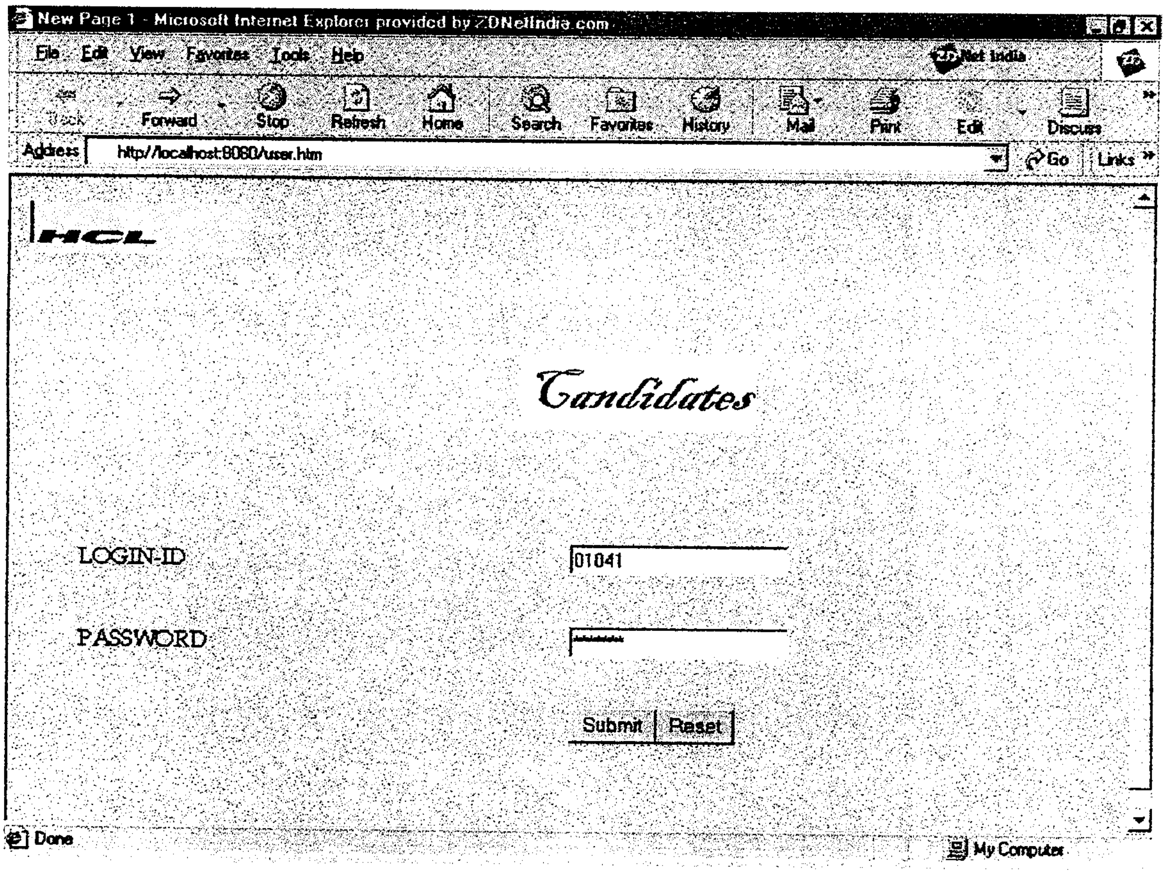
APPENDIX-B


New Page 1 - Microsoft Internet Explorer provided by ZDNetIndia.com
File Edit View Favorites Tools Help
Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:8080/recruitmentfirst.html
Go Links
HCL
Done
My Computer

RECRUITMENT SYSTEM

Candidates *Administrator*

FIRST PAGE

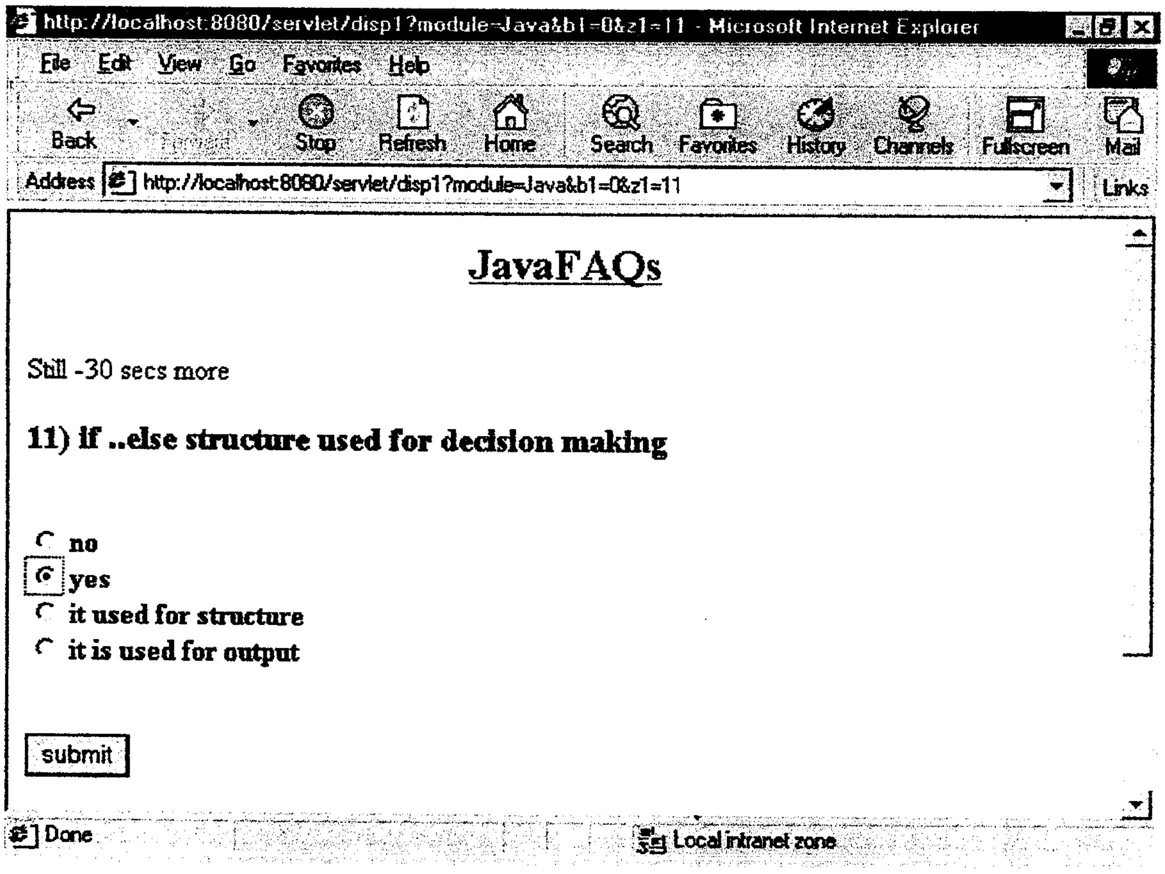

New Page 1 - Microsoft Internet Explorer provided by ZDNetIndia.com
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http//localhost:8080/user.htm
Go Links
HCL
Done
My Computer

Candidates

LOGIN-ID	01041
PASSWORD	******

Submit Reset

USER ENTER FOR EXAM


http://localhost:8080/servlet/disp1?module=Java&b1=0&z1=11 - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail
Address http://localhost:8080/servlet/disp1?module=Java&b1=0&z1=11
Links

JavaFAQs

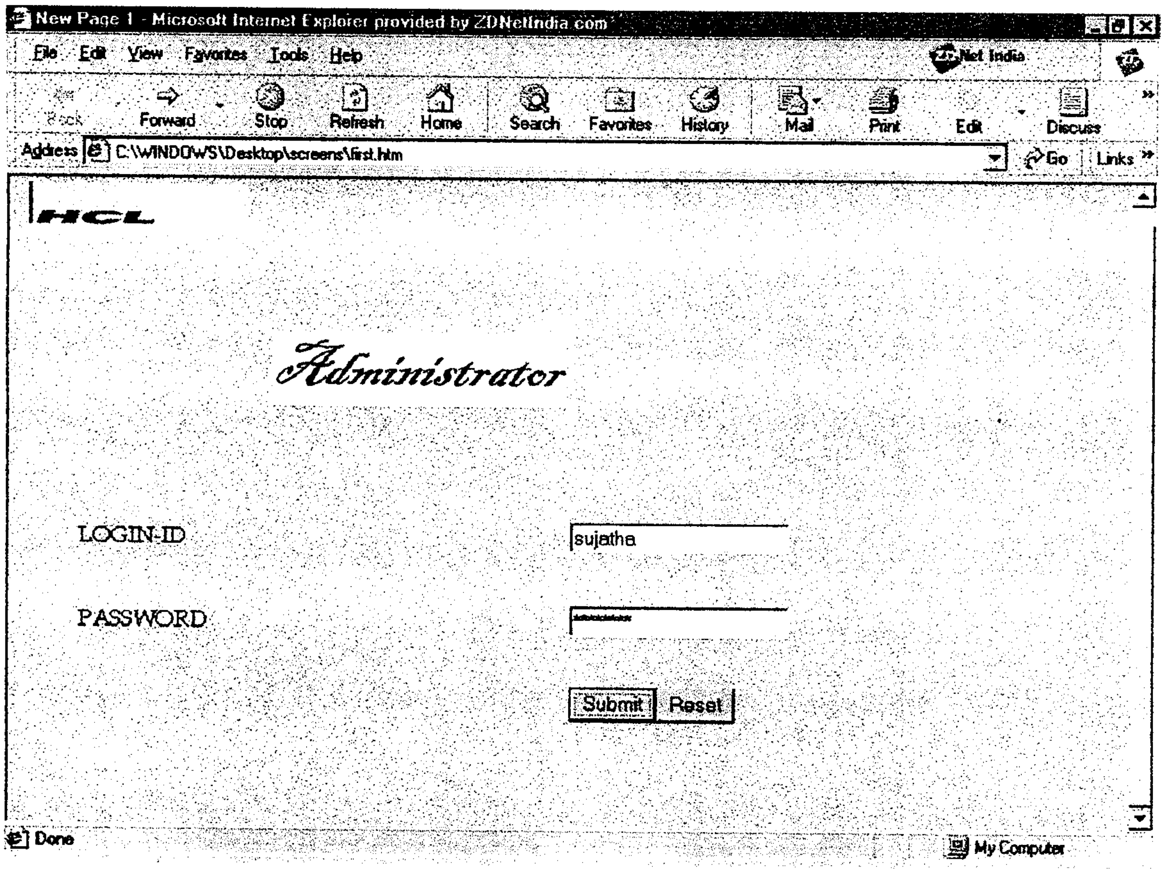
Still -30 secs more

11) if ..else structure used for decision making

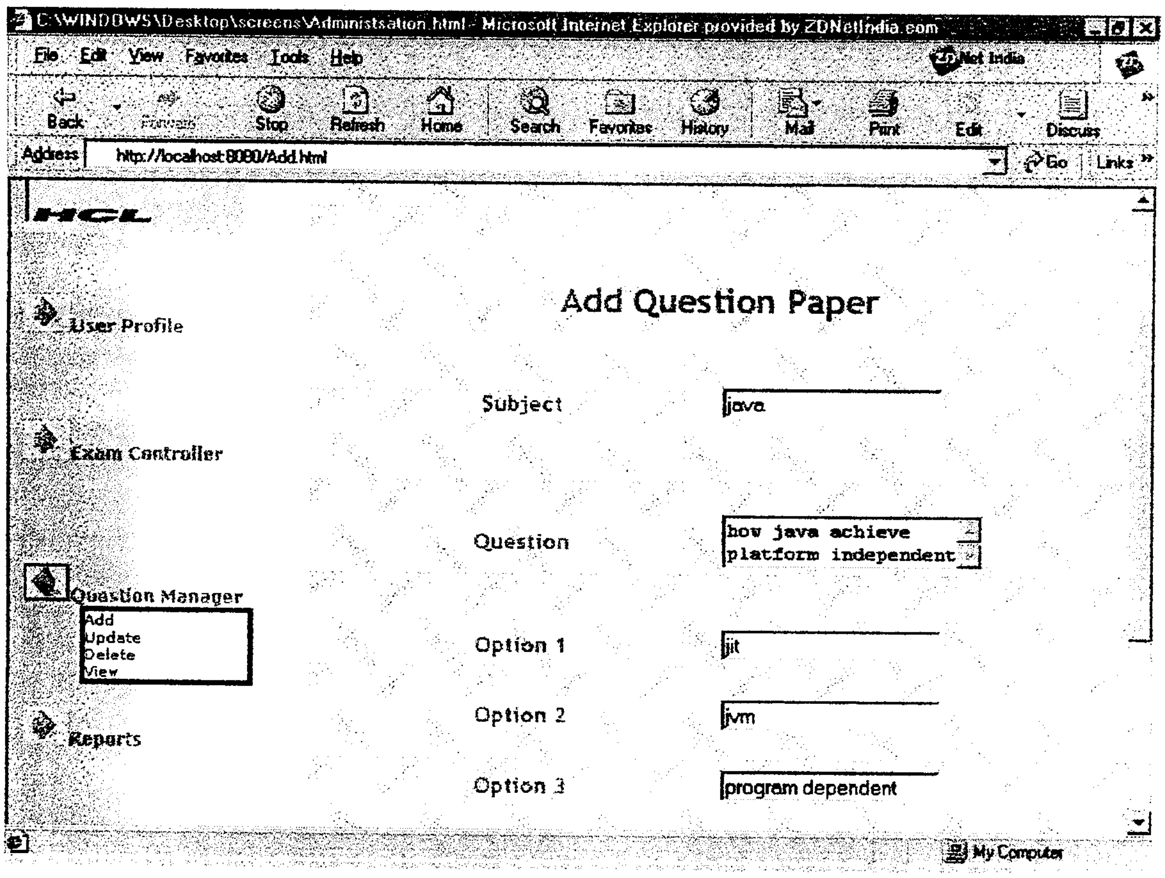
- no
- [✓ yes]
- it used for structure
- it is used for output

submit

QUESTIONS FROM QUESTIONBANK TO USER


New Page 1 - Microsoft Internet Explorer provided by ZDNetIndia.com
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address C:\WINDOWS\Desktop\screens\first.htm
Go Links
HCL
Administrator
LOGIN-ID
sujatha
PASSWORD
Submit
Reset
Done
My Computer

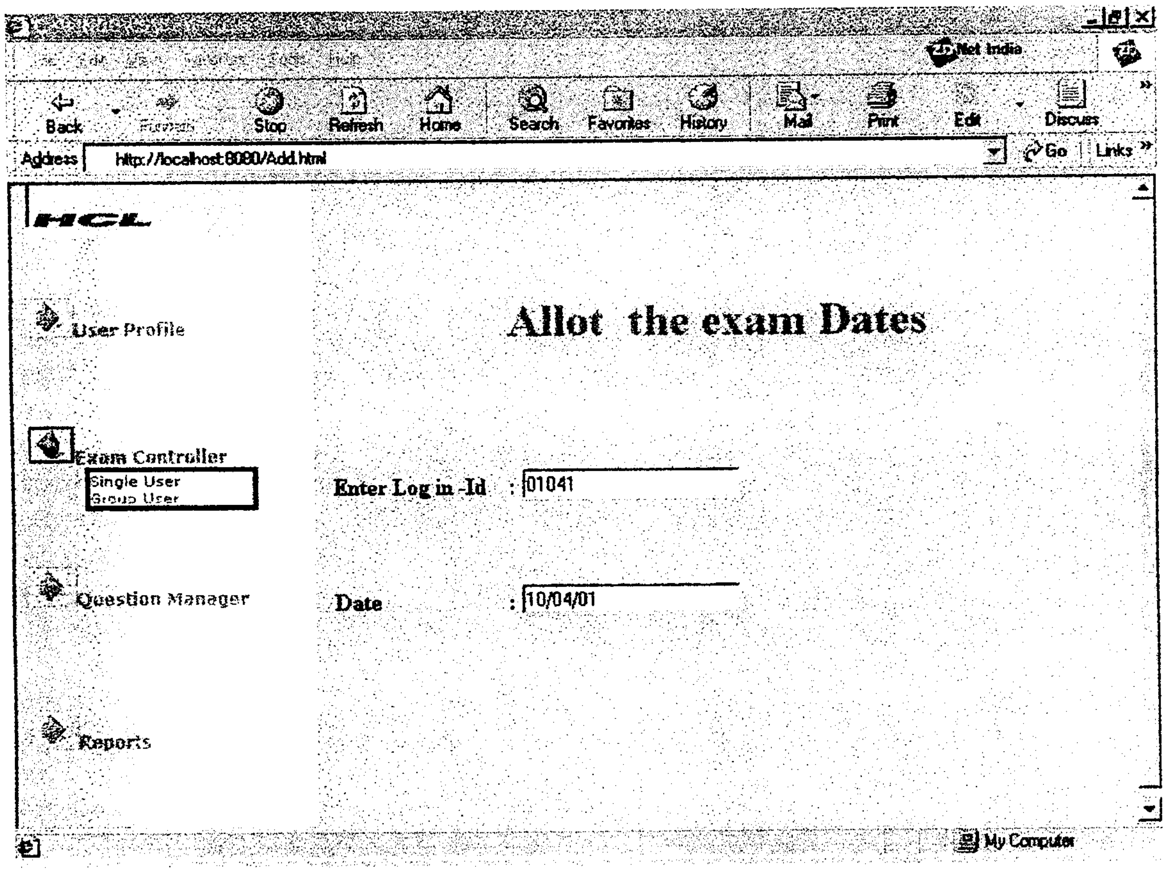
ADMIN-ENTER FOR OPERATION


C:\WINDOWS\Desktop\screens\Administsation.html - Microsoft Internet Explorer provided by ZDNetIndia.com
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:8080/Add.html
Go Links
HCL
User Profile
Exam Controller
Question Manager
Add
Update
Delete
View
Reports
My Computer

Add Question Paper

Subject	java
Question	how java achieve platform independent
Option 1	jit
Option 2	jvm
Option 3	program dependent

ADMIN-ADD QUESTIONS

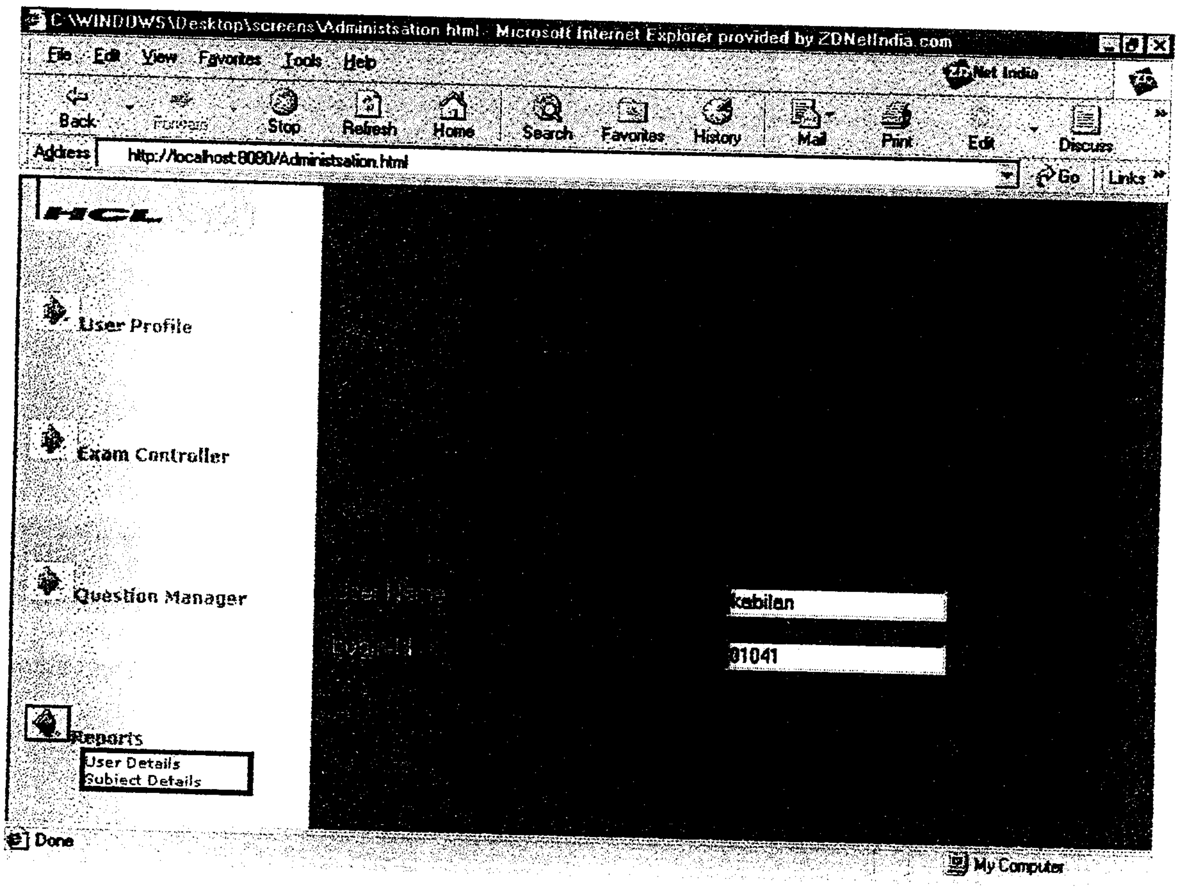


Allot the exam Dates

Enter Log in -Id : 01041

Date : 10/04/01

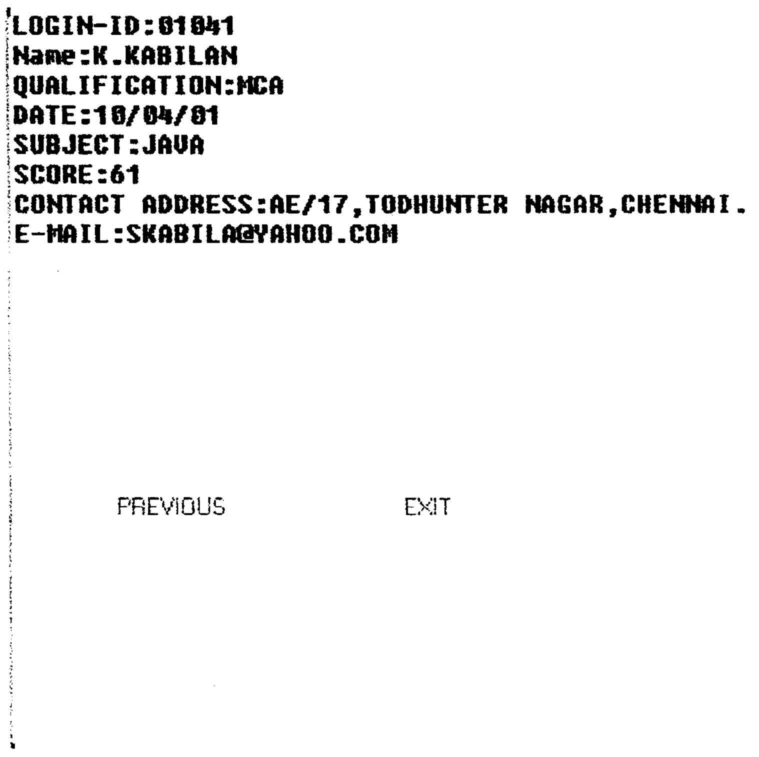
ADMIN-ALLOT EXAM DATE


C:\WINDOWS\Desktop\screens\Administsation.html - Microsoft Internet Explorer provided by ZDNetIndia.com
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:8080/Administsation.html
Go Links
HCL
User Profile
Exam Controller
Question Manager
Reports
User Details
Subject Details
Done
My Computer

User Name kabilan

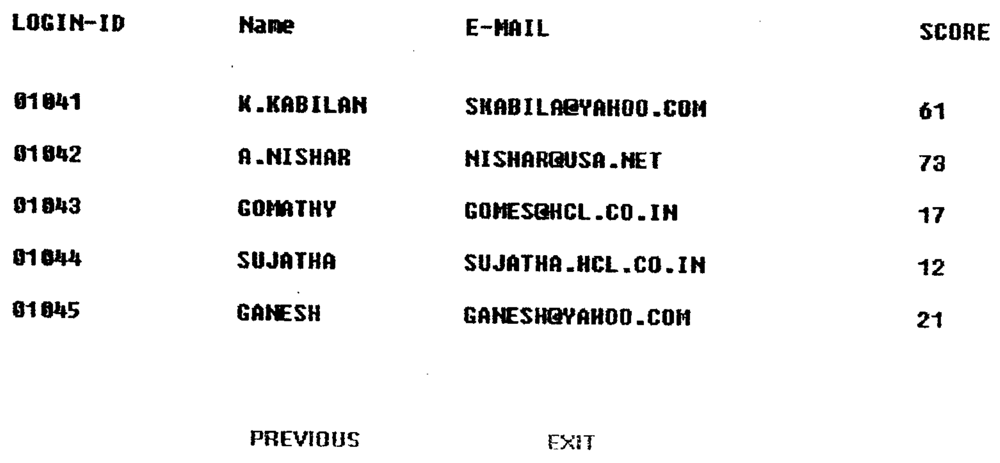
Login Id 01041

ADMIN-GET USER REPORTS



LOGIN-ID:01041
Name:K.KABILAN
QUALIFICATION:MCA
DATE:10/04/01
SUBJECT:JAVA
SCORE:61
CONTACT ADDRESS:AE/17,TODHUNTER NAGAR,CHENNAI.
E-MAIL:SKABILA@YAHOO.COM

REPORT FROM ADMIN-CANDIDATE REPORT



LOGIN-ID	Name	E-MAIL	SCORE
01041	K.KABILAN	SKABILA@YAHOO.COM	61
01042	A.NISHAR	NISHAR@USA.NET	73
01043	GOMATHY	GOMES@HCL.CO.IN	17
01044	SUJATHA	SUJATHA.HCL.CO.IN	12
01045	GANESH	GANESH@YAHOO.COM	21

REPORT FROM ADMIN-SUBJECT DETAILS