TITLE- DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION

Developed By:

1. Md. Akhtar Husain

2. Ashutosh Kumar

3. Irfan Ahmed

4. Vibha Kumari

Under the Supervision of:

Mr. Ashifuddin Mondal

Assistant Professor,

Department of Computer Science and Engineering

01.11.2013

TITLE- DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION

A Dissertation Submitted in partial fulfillment for the Degree of Bachelor of Technology (B.TECH), 7th Semester in Computer Science & Engineering.

Submitted By:

Name	University Registration Number	University Roll Number
Md Akhtar Husain	101270110047	12700110118
Ashutosh Kumar	101270110024	12700110025
Irfan Ahmed	101270110038	12700110017
Vibha Kumari	101270110116	12700110069

Under the Supervision of:

Mr. Ashifuddin Mondal

Assistant Professor,

Department of Computer Science and Engineering

West Bengal University OF Technology

Narula Institute of Technology

(November 2013)

CERTIFICATE OF APPROVAL

This is to certify that the project entitled “RESUME BUILDER APPLICATION” has been carried out by Md Akhtar Husain, Ashutosh Kumar, Irfan Ahmed and Vibha Kumari under my supervision in partial fulfillment for the degree of Bachelor of Technology (B.TECH) in Computer Science & Engineering of Narula Institute of Technology, Agarpara under West Bengal University of Technology during the academic year 2013-14.

It is understood that by this approval the undersigned do not necessarily endorse any of the statements made or opinion expressed therein but approves it only for the purpose for which it is submitted.

Submitted By:

Name	**University Roll No**	**University Registration No**
Md Akhtar Husain	12700110118	101270110047
Ashutosh Kumar	12700110118	101270110047
Irfan Ahmed	12700110118	101270110047
Vibha Kumari	12700110118	101270110047

-------------------------   ------------------------------

Ashifuddin Mondal,

Assistant Professor,

Jayanta Pal,

Head, CSE Dept

Department of Computer Science and Engineering

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide Mr. Ashifuddin Mondal for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

CONTENTS

ABSTRACT

The Project "**DEVELOPMENT OF A FUTURE RICH DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION**" develops an internet application for department of computer science for automating the process of resume writing. This would be facilitating the students to make and print their resumes in a proper format. In addition, it will be facilitating the higher management to search the students depending upon their skill sets and other attributes. The basic requirement is to have a centralized repository of all skill-holders in the organizations that a student with a particular skill set can immediately found in case of urgent requirement.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Project title as "**DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION"** is a web based Application aimed for managing the information i.e., educational, personal details of persons & professional details of freshers as well as experienced employees of Noble Organization.

1.2 Project Overview

This project **"DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION"** targeted for Noble Organizations is developed for the benefit of the users for providing the precise and correct information like number of resume formats, educational details, personals details applying for jobs etc.

The project is divided into four modules, they are:

- **User Registration**
- **Resume Preparation**
- **Resume types**
- **Reports**

User registration:

This module is used for new users. Already existing users can directly access the resume preparation, but in the case of new users they have to use the user registration module.

Resume preparation:

This module is used by the existing and registered users to prepare their resumes in proper format and retrieve the resume details**.**

Resume types:

This module is used by the existing users to directly print the give format of resume.

Reports:

This module is used by any higher management to search students on the basis of their skills.

CHAPTER 2

PROBLEM DEFINITION

The Project “**DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION**” is an intranet application for the department of computer science so as to automate the process of resume writing & applying for jobs. This would be facilitating the students & experienced candidates to make and print their resumes in a proper format. In addition, it will be facilitating the higher management to search the students depending upon their skill sets and other attributes. The basic requirement is to have a centralized repository of all skill-holders in the organizations that a student with a particular skill set can immediately found in the case of urgent requirement. And also searching for jobs &applying for jobs is possible.

2.1 Existing System

The existing system of the resume builder is prepared in the MS-Word Application software. The Format which is designed is manual. But creating different resume formats is not easy. To reduce the burden we have developed this software.

The different processes involved are:

- ✓ To maintain the details like personal, educational &skills.
- ✓ To maintain the three default resume formats.
- ✓ Select appropriate one and display the data in required format.

2.2 Drawbacks of the existing system

The existing system has lot of problems such as

- ✓ There is no database to store and retrieve the details from MS-Word
- ✓ Time delay is more because we have to prepare the resume manually.

2.3 Proposed System

The proposed system tries to solve the problems mentioned above. The main objective of the proposed system is to provide information instantly as and when it is required. The main objective is to make the DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION details more efficient. This system should maintain different data files and resume formats, so that the data can be retrieved easily and in an efficient manner. The system is very interactive. It should ensure process integration to the desired extent, various reports should be generated as the need be. This system should also ensure that there is no redundancy in the recorded data.

2.4 Advantages of proposed system:

1. To store all details of the persons i.e. personal, educational, and skill details, so that it provides better services to users.
2. The proposed system provides efficient management of resumes.
3. Because of computerization manual work is eliminated and manual errors can be overcome easily.

2.5 Limitations of the proposed system

1. The main drawback of the proposed system is that, it is not provided with any help menus.
2. The proposed system does not provide the reports.

2.6 Software Requirement Specifications

Software Requirements:

Operating System	:	Windows 2000/xp-sp2
Back - End	:	ORACLE9i
Front - End	:	JSP
Browser	:	IE 7

Hardware Requirements

CPU	: Pentium 4, 1.7 GHZ
RAM	: 512Mb
Hard disc	: 80 GB

Number of persons

The project titled as **“DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION”** is handled by **three** persons lead by a project in charge.

CHAPTER 3

Feasibility Study

All projects are feasible – given unlimited resources and infinite time! Unfortunately, the development of computer-based system or product is more likely plagued by a scarcity of resources and difficult to generate default resume formats. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Months or years of effort, thousands or millions of dollars, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase.

Generally the feasibility study is used for determining the resource requirement cost, benefits and whether the proposed system is feasible with respect to the organization .The feasibility of proposed Automation of Resume Builder for Noble College of Computer Sciences could be evaluated as follows. There are three types of feasibility which are equally important. They are

- Economical feasibility
- Technical feasibility
- Operational feasibility

3.1 Economical Feasibility

Economical feasibility is concerned with cost savings, increased profits & reductions in efforts. It shows how much beneficial is the new developed system over the existing system.

System startup cost

PC with Pentium-4 Processor	Rs. 25,000/-
Line Printer	Rs. 6,500/-
Furniture Indirect cost for site preparation	Rs. 3,500/-
Purchase of Software	Rs. 10,000/-

	Rs. 45, 000/-

System operating costs

- Additional equipment (CDs, ribbons, power supply) maintenances
- Program maintenance
- Stationary

Total operating cost	Rs. 5,000/-

Total Cost of the Implementation of	
The Proposed System	Rs. 45,000/-
	Rs. 5,000/-

	Rs. 50,000/-

As this project is related to service sector which aims at providing better service to all persons and does not yield any income. Providing better service is the basic requirement of any organization either service sector or financial sector. So the proposed system is economically feasible as it also reduces manpower utilization.

3.2 Technical Feasibility

Technical feasibility deals with the existing technology, software and hardware requirements of the proposed system. The proposed system **“DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION for Noble College of Computer Sciences”** needs the following:

-Personal computer with a Pentium-4 processor, 512 MB RAM.
-Line printer
-ORACLE for DB backend
- JSP as front-end tool

In order to implement the proposed system necessary technology will be acquired. Hence the proposed system is technically feasible.

3.3 Operational Feasibility

Operational feasibility is the willingness & ability of the management, Employees, Students and others to use & support a proposed system. As concerned to the resume builder of Noble College of Computer Sciences, all the staff is in desperate need of a *Computer Based Information System* (CBIS) to reduce the manual effort & for accurate information. There is no difficulty in handling the system. There is full support from Management. So the system is operationally feasible. The proposed *Computer Based* Information *System* (CBIS) for Noble College of Computer Sciences is in no way inferior to the existing manual system and it yields better results than the present manual system. This system can give good support and makes the services easy.

CHAPTER 4

SYSTEM REQUIREMENT ANALYSIS

System analysis is an important activity that takes place when we are building a new information system or changing existing ones, analysis is used to gain an understanding of an existing system and what is required of it. At the conclusion of analysis, there is system description and set of requirements for a new system. If there is no existing system, the analysis defines only the requirements.

System models are used to gain precision and to avoid the ambiguities often found in the natural language system descriptions modeling techniques used in the system analysis avoids ambiguity by using precise modeling constructors and process descriptions. They also assist analysts to define precisely the requirement of the new system. Software tools that help analyst in their work now often support system analysis. These tolls are the models developed during analysis and some convert these models to trail designs.

This phase is detailed appraisal of the existing system. The appraisal includes finding how the system works and what it does. It also includes system's problems and what the end-users required for any new or changed system. After this phase, analyst should be familiar with both the detailed operation of the system and what is required of the new system.

Analysts must spend considerable time in examining components of exiting system. Analysis has to find out what information is send between the end – users and the staff.

One of the most important factors in system analysis is to understand the system and its problems. A good understanding of the system enables designers to identify and correct problems and suggest realistic solutions for them. It also helps them to develop solutions that satisfy all users and thus make the new system acceptable in an organization. System users are the first information source investigated by the analysts.

Form users it is to find out the existing system activities and to develop the user’s objectives and requirements. a system analyst must spend a lot of time talking to users and finding how they use the system, Any problem they find with the system and what they expect from it.

4.1 Software Development Life Cycle

The sequence of events in the development of an information system (application), which requires mutual effort on the part of user and technical staff is called Software Development Life Cycle.

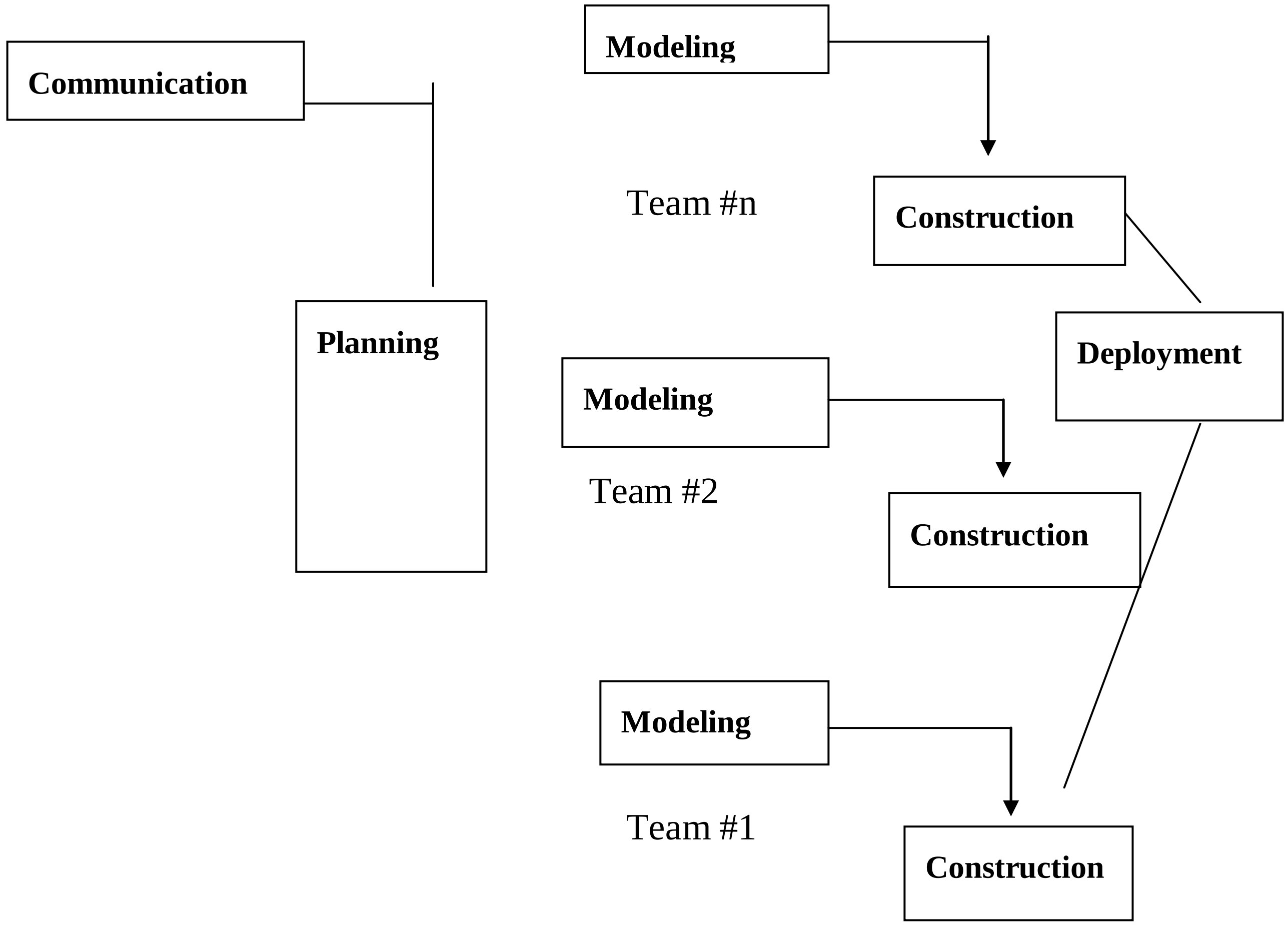
Process Models

RAD Model

Rapid Application Development (RAD) is an incremental software process model that emphasizes a short development cycle. RAD model is a high speed adoption of the waterfall model, in which rapid development is achieved by using a component based construction approach. If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a fully functional system with in a very short period.

In RAD model, communication works to understand the business problem and information characteristics. Planning is essential because multiple software teams work in parallel on different system functions. Modeling establishes design representations that serve as the basis for RAD construction activity. Construction emphasizes the use of pre existing software components. Finally the deployment establishes a basis for subsequent iterations. The RAD process model is illustrated in the figure below.

Diagram:



4.2 Entity-Relationship Diagram (ERD)

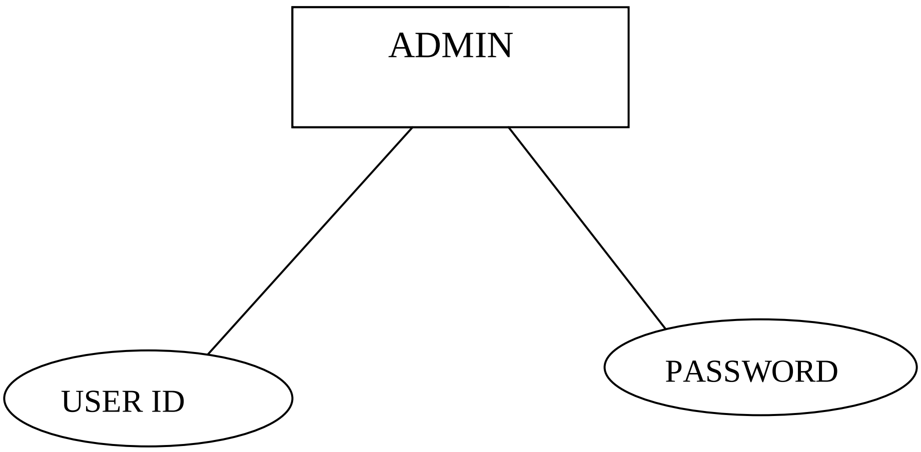
The Entity-Relationship Diagram depicts a relationship between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description.

At first a set of primary components are identified for ERD i.e. Data objects, attributes relationships and various type indicators. Data objects are represented by labeled rectangles. Relationships are indicated with labeled lines connecting objects. Connectors between data objects and relationships are established using a variety of special symbols that indicate cardinality and modality. ERD notation also provides a mechanism that represents the association between objects.

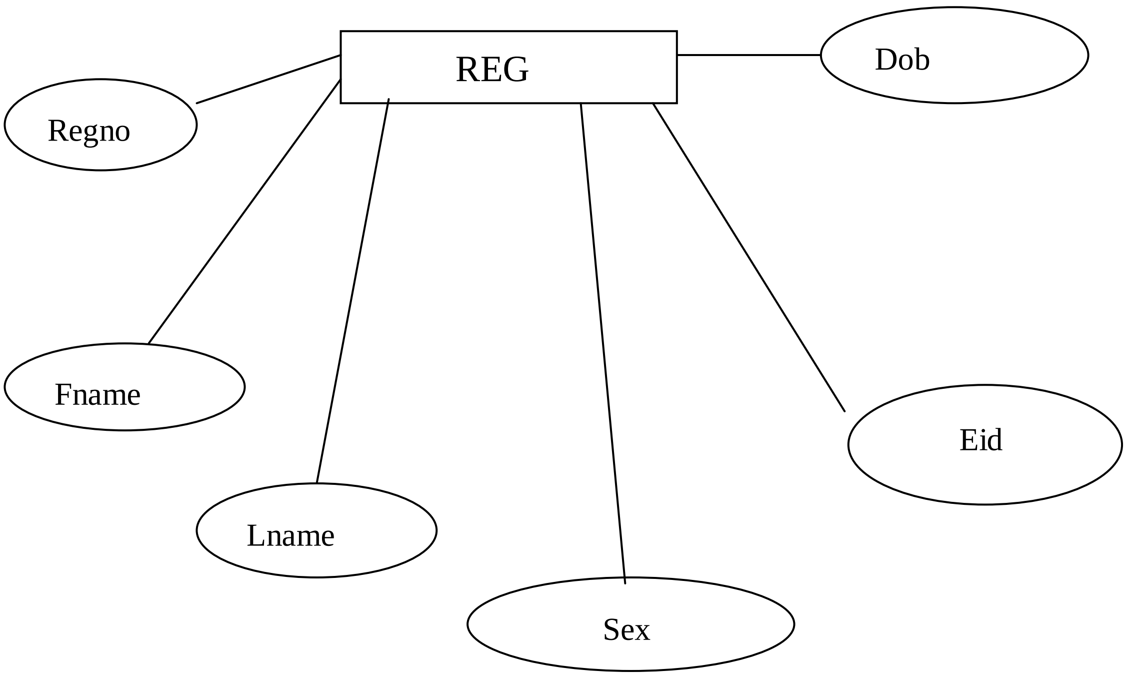
Data modeling and the entity- relationship diagram provides the analyst with a concise notation for examining data within the context of data processing application.

The ER Diagram for the overall proposed system is given in the figures

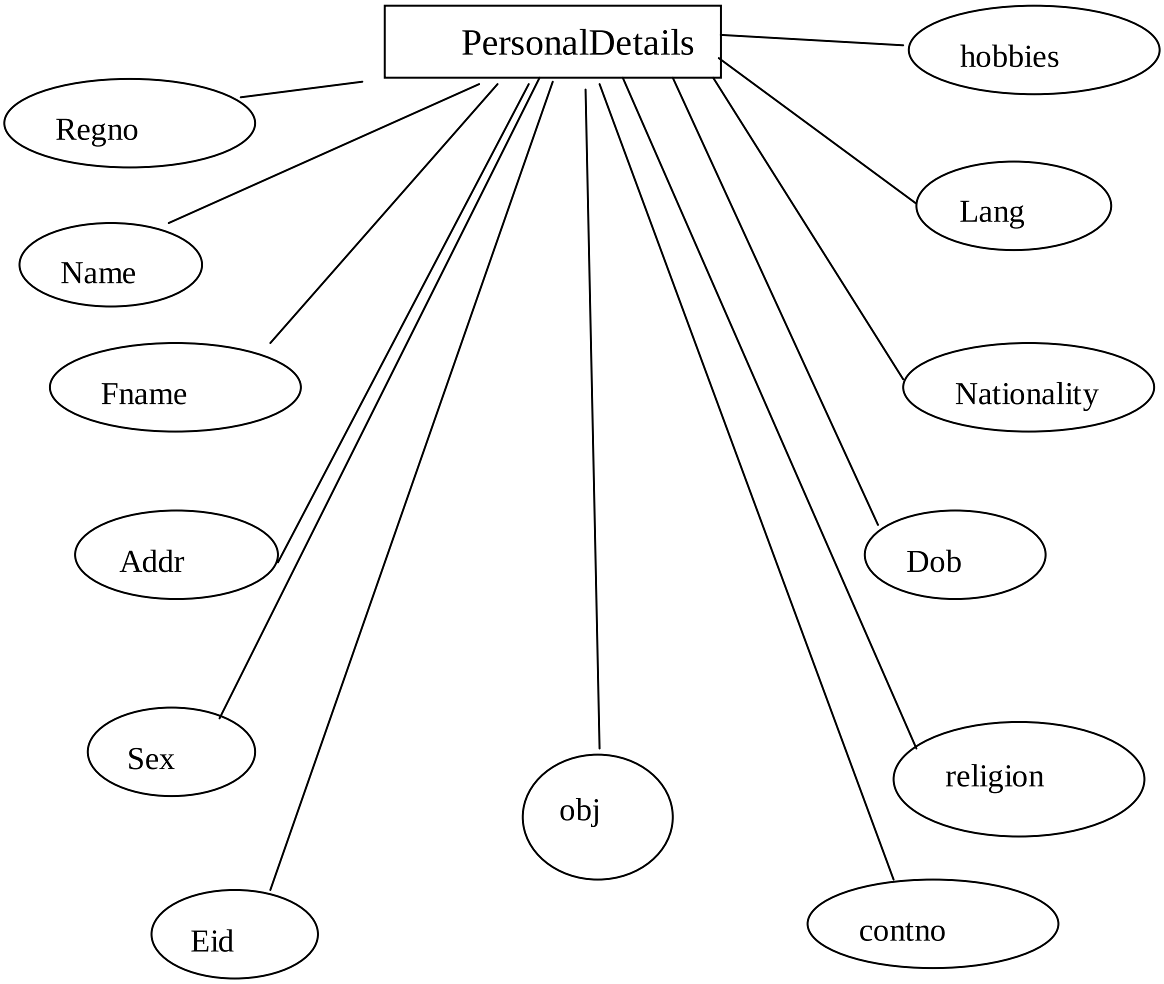
ADMIN:



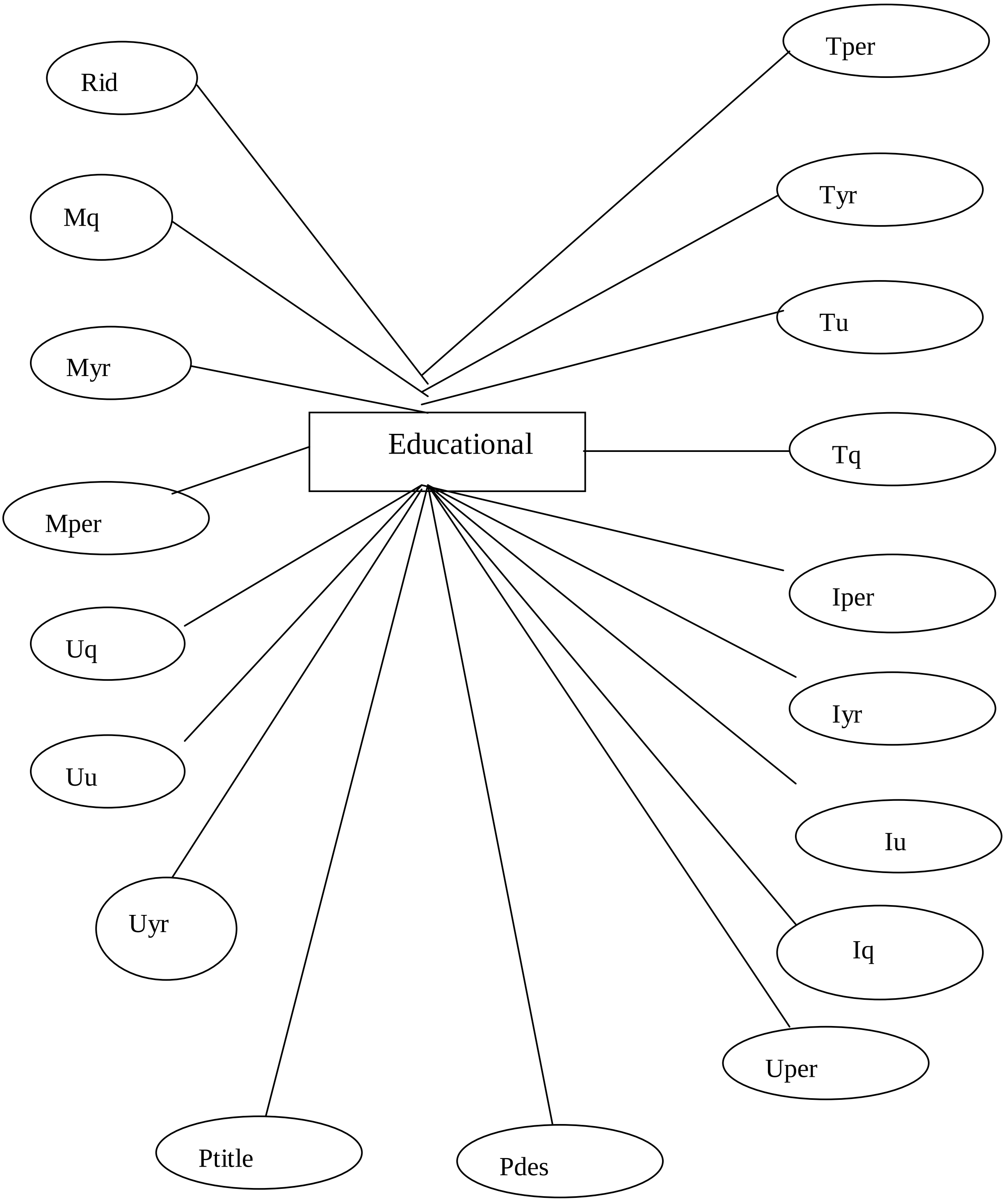
Registration:



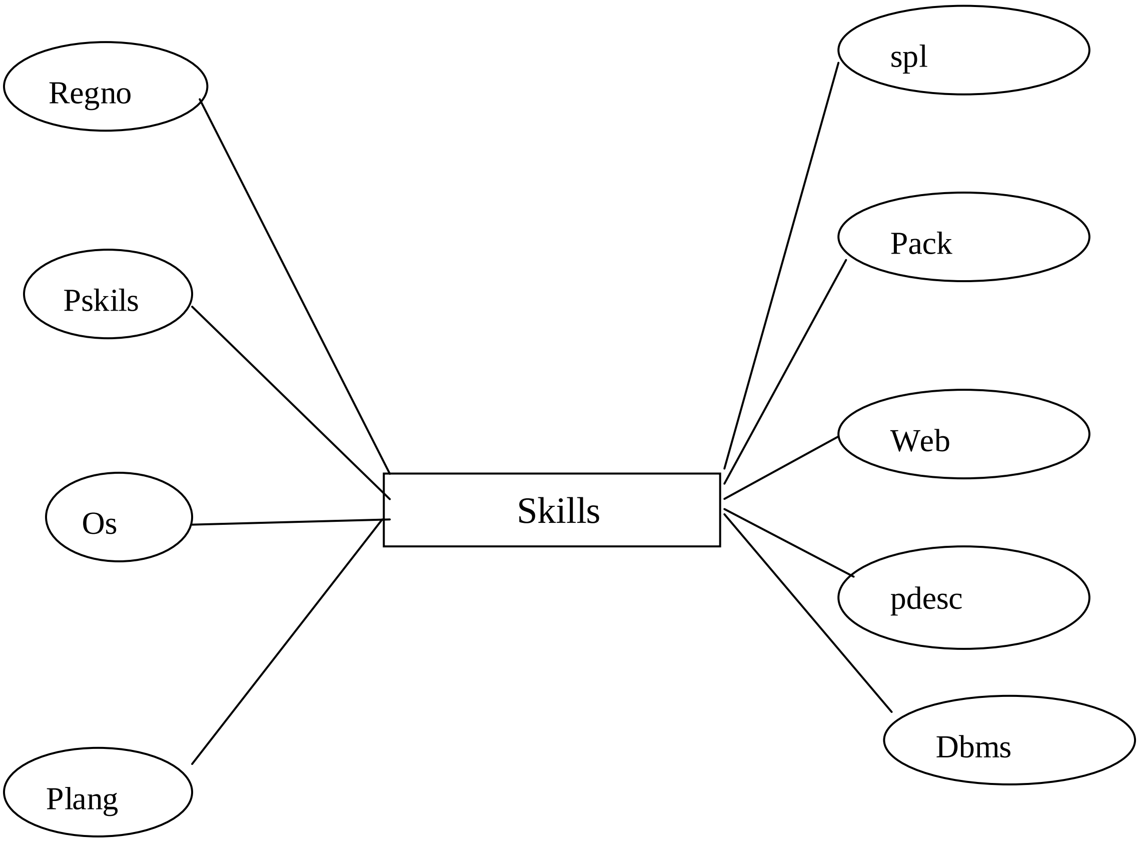
Personal Details:



Educational details:



Skills:



4.3 Data Flow Diagram (DFD)

The data flow diagram (DFD) is one the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and those transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD s may be partitioned into levels that represent increasing information flow and functional details.

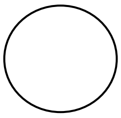
A level 0 DFD, also called as the context diagram, represents the entire system as a single module with input and output data indicated by incoming and outgoing arrows respectively. Additional process and information flow paths are represented, as the level 0 DFD is partitioned to reveal more details.

A level 1 DFD, also called as top-level DFD, represent the system with major modules and data stores. The other levels will show each module in the top-level DFD in a more detailed fashion.

Notation:

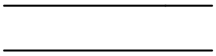
Data Flow: Data move in a specific direction from an origin to a destination. The data flow is a “packet” of data.

Process: People, procedures or devices that produce data. The physical component is not identified.



Source or Destination of data: External sources or destinations of data, which may be people or organizations or other entities.

Data Store: Here, the data referred by a process in the system.



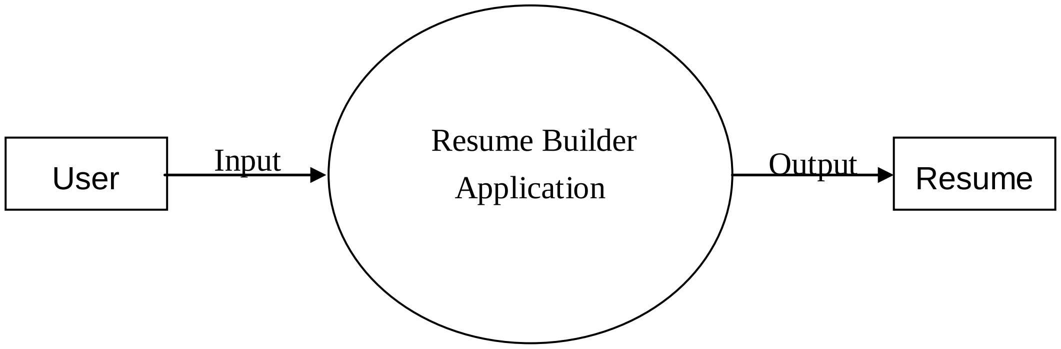
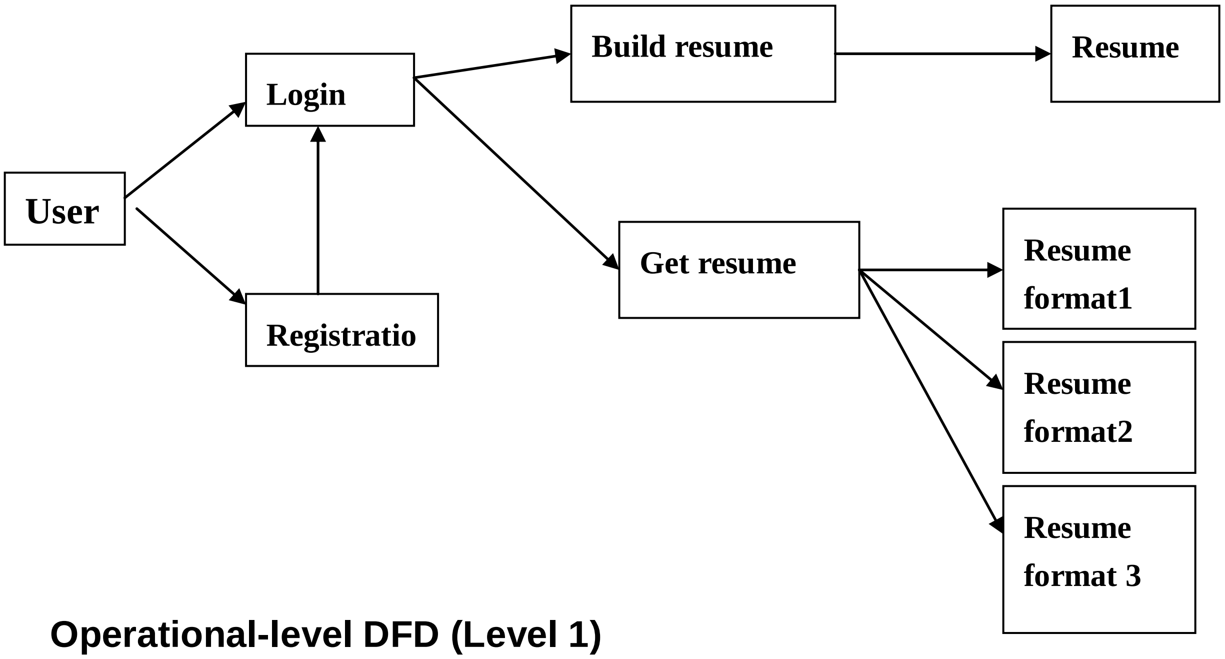


Fig 4.2 Context-level DFD (Level 0)

RESUME BUILDER (Data Flow Diagram)



Operational-level DFD (Level 1)

4.4 Data Dictionary

A data dictionary is a catalog – a repository – of the elements in the system. As the name suggests, these elements center on data and the way they are structured to meet user requirements and organization needs. A data dictionary is a collection of data about data. The basic idea is to provide information on the definition, structure and the user of the element an organization uses. The data element is a unit of data that can't be decomposed.

Tables:

Table Name: Admin Table
Description: to store the user id & passwords

Field Name	Data Type	Size	Constraints
Userid	Varchar2	20	Primary key
Password	Varchar2	20	

Table Name: Registration Table
Description: to store all the details of the person

Field Name	Data Type	Size	Constraints
Regno	Number	5	PrimaryKey
Fname	Varchar2	20	
Lname	Varchar2	20	
Gender	Varchar2	10	
Dob	Date/Time	15	
Eid	Varchar2	30	
Userid	Varchar2	25	
Password	Varchar2	25	

Table Name: Skills Table
Description: To store all details of the employee

Field Name	Data Type	Size	constraint
Regno	Number	4	PrimaryKey
Pskills	Varchar2	200	
Os	Varchar2	30	
Plang	Varchar2	40	
Dbms	Varchar2	30	
Web	Varchar2	30	
Pack	Varchar2	30	
Spl	Varchar2	20	
Declar	Varchar2	50	

Table Name: Personal details Table
Description: To store all details of the persons

Field Name	**Data Type**	**Size**	**Constraints**
Regno	Number	4	Primary key
Fname	**Varchar2**	20	
Lname	**Varchar2**	10	
Adder	**Varchar2**	100	
Gender	**Varchar2**	9	
Dob	Date/time		
Nationality	**Varchar2**	15	
Religion	**Varchar2**	15	
Hobbies	**Varchar2**	100	
Languages	**Varchar2**	30	
Email	**Varchar2**	30	
Contno	Number	20	
Obj	**Varchar2**	200	

Table Name: Educational Details Table
Description: To store all details of the employee

Field name	**Data Type**	**Size**	**Constraint**
Regno	Number	5	Primary key
Mqual	**Varchar2**	15	
Mdur	**Varchar2**	15	
Muni	**Varchar2**	8	
Mper	**Varchar2**	4	
Bqual	**Varchar2**	10	
Bdur	**Varchar2**	20	
Buni	**Varchar2**	8	
Bper	**Varchar2**	4	
Ssc	**Varchar2**	15	
Sdur	**Varchar2**	15	
Sedu	**Varchar2**	8	
Sper	**Varchar2**	4	
Exp	**Varchar2**	**30**	
Ptitle	**Varchar2**	10	
Pdes	**Varchar2**	8	

Chapter 5

System Design

Introduction

Design is the first step in the development phase for any engineering product (or) system. It may be defined as " the process of applying various techniques and principles for the purpose of defining a device, a process, or a system insufficient detail to permit its physical realization".

Software design is an iterative process through which requirements are translated into a 'Blue print' for constructing the software. The design is represented at a high level of abstraction, a level that can be directly translated to specific data, functional and behavioral requirements.

Preliminary design is concerned with the transformation of requirements into a data and software architecture. Detail design focuses on refinements to the architectural representation. That leads to detailed Data structure and algorithmic representation for software.

In the design step, the element of the analysis model gets converted into a data design, and architectural design, an interface design and a procedural design.

The data design transforms the information domain model created during analysis into the data structures that will be required to implement software.

The architectural design defines the relationship among major structural elements of the program.

The interface design describes how the software communicates within itself, to systems that interoperate with it, and with humans who use it. An interface implies a flow of information (e.g., data and /pr control). Therefore, the data and control flow diagrams provide the information required for interface design.

Design Process

Design process is in between the analysis and implementation process. The following design diagrams (data flow diagrams and E-R diagrams) make it easy to understand and implement. The design process of software system has two levels.

1. Systems Design or Top Level Design.
2. Detailed Design or Logical Design.

System Design or Top Level Design:

In the system design the focus is on deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

Detailed Design or Logical Design:

In detailed design the interconnection of the modules or how the specifications of the modules can be satisfied is decided . Some properties for a software system design are

- Verifiability.
- Completeness.
- Consistency.
- Trace ability.
- Simplicity/understandability.

Design principles:

Basic design principles that enable the software engineer to navigate the design process are.

- ➢ The design process should not suffer from “Tunnel vision”.
- ➢ The design should be traceable to the analysis model.
- ➢ The design should not reinvent the wheel.
- ➢ The design should minimize the intellectual distance between the Software and the problem, as it exists in the real would.
- ➢ The design should exhibit uniformity and integrity.
- ➢ The design should be structured to accommodate changes.
- ➢ The design is not coding. The coding is not a design.
- ➢ The design should be assessed for the quality, as it is being Create, not after the fact.
- ➢ The design should be reviewed to minimize the conceptual errors.

Database Design:

The goal of Database Design is to generate a set of relation schemes that allow us to store information without unnecessary redundancy and allows us to retrieve information easily. We can achieve optimization, ease of use in maintenance by designing the database using relational model between or among the tables.

- To reduce redundancy.
- To arrive at loss-less join.
- To reduce the time as compared to the present system.
- To reduce the number of errors

5.5 Normalization:

Normalization of relation schema is done to eliminate insertion and deletion anomalies that exist in database.

Normalization is a step-by–step reversible process of converting given collection of relations to some more desirable form in which the relations have a progressively simpler and more regular structure. No information is lost in normalization process.

The objectives of Normalization are

- To make it feasible to represent any relation in the database.
- To obtain powerful retrieval algorithms based on a simpler collection of relational operations than would otherwise be necessary.
- To free relations form undesirable insertions, update and deletion Dependencies.
- To make the collection of relations neutral to query Statistics where these statistics are liable to changes as time goes by.

A relation R is said to be in 1NF if all underlying domains contain atomic values only.

A relation R is said to be in 2NF if and only if it is in 1NF and every non-key attribute is non-transitively dependent on the primary key. A relationship is said to be in 3NF if and only if it does not feature any non-trivial functional dependencies between non-prime attributes. A non-prime attribute is one that does not belong to any candidate key.

All the database tables like Login, order _Entry etc., used in the project have atomic values. For Example the Login table consists of Uname and pwd attributes and all the attributes are atomic values.

In the above table all the fields contain atomic values as no field has more than one value. So form the example it is clear that all underlying domains contain atomic values. So the first normal form is satisfied.

CHAPTER 6

Software Implementation

6.1 **Introduction**:

Implementation is the process of assuring that the information system is operational and then allowing users take over it s operation for use and evaluation. Implementation includes the following activities.

- ✓ Obtaining and installing the system hardware.
- ✓ Installing the system and making it run on its intended hardware.
- ✓ Providing user access to the system.
- ✓ Creating and updating the database.
- ✓ Training the users on the new system.
- ✓ Documenting the system for its users and for those who will be responsible for maintaining it in the future.
- ✓ Making arrangements to support the users as the system is used.
- ✓ Transferring on going responsibility for the system from its developers to the operations or maintenance part.
- ✓ Evaluating the operation and use of the system.

The most visible component of implementation process is the system conversion. The four basic conversion strategies include.

- Direct Conversion
- Parallel Conversion
- Pilot conversion
- Phased Conversion

6.1.1 Direct Conversion

In direct conversion, the organization stops using the old system and starts using the new one at the same time.

6.1.2 Parallel Conversion

A parallel conversion involves running both old system and new system and comparing their results. The new system is accepted only after the results have matched for an acceptable period.

6.1.3 Pilot Conversion

Pilot conversion means introducing the new system to a small part of the organization, expanding its use once it is known to operate property there. Eventually, it will be in use by the entire organization.

6.1.4 Phased Conversion

Phased conversion means introducing a system in stages, one component or module at a time, waiting until that one is operating properly before introducing the next.

6.2 Hardware Environment

CPU	: Pentium 4, 1.7 GHz
RAM	: 512 MB
Hard disk	: 80 GB

6.3 Software Environment

Operating Environment	: Windows200/XP
Back-End	: ORACLE
Front End	: JSP
Browser	: IE7

CHAPTER 7

System Testing

7.1 Introduction

System testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Software testing fundamentals define the overriding objectives for software testing. Testing is one of the steps in the software engineering process that could be viewed (psychologically, at least) as destructive rather than constructive.

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

7.2 White-Box Testing (GLASS-BOX TESTING)

White-box tests focus on the program control structures. Test cases are derived to ensure that all statement in the program has been executed at least once during testing and that all logical conditions have been exercised.

Knowing the internal working of a product, tests can be conducted to ensure that " all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised.

White box testing is a test case design methods that uses the control structure of the procedural design to design to derive test cases.

- ❖ Guarantee that all independent paths within a module have been exercised at least once.
- ❖ Exercise all logical decision on their and false sides
- ❖ Execute all loops at their boundaries and within their operational bounds.
- ❖ Exercise internal data structures to ensure their validity.

7.3 Black-Box Testing:

Black-Box testing focuses on the functional requirements of the software .it is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

Black- box testing attempts to find the following types of Errors

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Performance errors
- Initialization and termination errors.

In white-box testing, test cases are performing early n the testing processes where as in black-box testing process. It is applied during later stages of testing as the attention is focused on the information domain.

7.4 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design. Using the procedural design description, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of test and uncovered errors is limited by the constrained scope established for unit testing. The unit test is normally the constrained scope established for unit testing. The unit test is normally white box oriented, and the step can be conducted in parallel for multiple Modules.

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

7.5 Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associate with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. There is often a tendency to attempt non-incremental integration to construct the program using big bang approach. All modules are combined in advance. The entire program is tested as a whole.

Incremental integration is the anti thesis of big bang approach. The program is constructed and tested in small segments where errors are easier to isolate and correct; interfaces are more likely to be tested completely ; and systematic test applied. Modules are integrated by moving down word through the control hierarchy, beginning with the main control module.

7.6 Verification and Validation:

Validation is a process of finding out if the product being built is right? I.e. whatever the software product is being developed; it should do what the user expects it to do. The software product should functionally do
What it is supposed to, it should satisfy all the functional requirements set by the user. Validation is done during or at the end of the development process in order to determine whether the product satisfies specified requirements.

The standard definition of Verification goes like this: “are we building the product RIGHT?” i.e. verification is a process that makes it sure that the software product is developed the right way. the software should confirm to its predefined specifications, as the product development agoes through different stages, an analysis is done to ensure that all required specifications are meet.

Validation and verification process go hand in hand. But visibly validation process starts after verification process ends (after coding of the product ends). Each verification activity (such as requirement specification, verification functional design verification etc.) has its corresponding validation activity (such as functional validation/testing, code validation/testing, system/integration validation,etc).

CHAPTER 8

Literature Survey

8.1 About Java:

8.1.1 Overview of java Technology:

Java has had a major impact on the computing scene. When the history of computers is written, its name will be up there with the stars.

If you were to choose just one language to learn today, it should be java. It's being pushed aggressively by sun and is growing by leaps and bounds. There are lots of java programmers out there and more join the party every day.

Java started out as a bit of an accident. A team under Bill Joy was working at sun on a new programming language for embedded applications. Java was originally expected to work in toasters and fridges, not on modern computers! the initial prognosis for java was not good and it was only the rise of the internet which saved java from oblivion. Since then, neither the Net nor Sun nor java has looked back and all have grown from strength to strength.

World Wide Web is an open ended information retrieval system- designed to be used in the distributed environment. This system contains web pages that provide both information and controls. We can navigate to a new web page in any direction. This is made- possible worth HTML java was meant to be used in distributed environment such as internet. So java could be easily incorporated into the web system and is capable of supporting animation graphics, games and other special effect. The web has become more dynamic and interactive with support of java. We can run a java program on remote machine over internet with the support of web.

8.1.2 Java Environment

Java environment includes a large number of tools which are part of the system known as java development kit(JDK) and hundreds of classes, methods, and interfaces grouped onto packages forms part of java standard library(JSL);

Java Architecture:

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the java virtual machine which are then interpreted on each platform by the runtime environment .java also provides stringent compile and runtime checking and automatic memory management in order to ensure solid code.

Java Virtual Machine

When we compile the code, java compiler creates machine code (byte codes) for a hypothetical machine called java virtual machine (JVM). The JVM will execute the byte code and overcomes the issue of portability. The code is written and compile for one machine and interpreted all other machines .this machine is called java virtual machine.

Paradigm of java

- Dynamic down loading applets(small application programs);
- Elimination of flatware phenomenon that is providing those features of a product that user needs at a time. The remaining features of a product can remain in the server.
- Changing economic model of the software
- Up-to-date software availability
- Supports network entire computing
- Supports CORBA&DCOM

8.1.3 An Overview of JSP

The java server pages TM Technology:

Java server pages TM technology is the java TM technology in the J2EE platform for building applications containing dynamic Web content such as HTML, DHTML, XJHTML and XML. The java Server pages technology enables the authoring of Web pages that create dynamic content easily but with maximum power and flexibility.

The java Server pages technology provides a textual description for the creation of response form request. The technology builds on the following concepts:

Template Data:

Substantial portions of dynamic content are actually fixed. The JSP technology allow for the natural manipulation of this data

Addition of Dynamic Data:

The JSP technology allows the addition of dynamic data to the template data in a way that is sample yet powerful.

Encapsulation of Functionality:

The JSP technology provides two related mechanisms for the encapsulation of functionality the standard java beans component architecture and the tag library mechanism.

Good Tool support:

The JSP technology has features that enable the creation of good authoring tools. The result is a flexible and powerful server-side technology.

Benefits of the Java Server Pages Technology:

The java Server pages technology offers a number of benefits:

- Write Once, Run Anywhere ^{TM} properties
- The java Sever pages technology is platform independent, both in its dynamic Web pages, Web servers, and its underlying server components. You can author JSP pages on any platform, run them on any web Server or web enabled application server, and access them form any web browser.
-

High quality tool support:

The Write Once, Run Anywhere properties of JSP allows the user to choose best- of – breed tools. Additionally an explicit goal of the java Server pages design is to enable the creation of high quality portable tools.

Separation of Roles:

JSP supports the separation of roles: developers write components that interact with server-side objects.

Reuse of components and tag libraries :

The java Server pages technology emphasizes the use of reusable components such as java Beans TM components. Enterprise java Beans TM
Components and tag libraries.

Separation of dynamic and static content :

The java Server pages technology enables the separation of static content form dynamic content that is inserted into the static template.

Support for scripting and actions :

The java server pages technology supports scripting elements as well as actions. Actions permit the encapsulation of useful functionality in a convenient form that can also be manipulated by tools. Scripts provide a mechanism to glue together this functionality in a per-page manner. Web access layer of N-tier enterprise application architecture(s). The Java Server pages technology is an integral part of the java 2 Platform Enterprise Edition (J2EE), which brings java technology to enterprise computing.

8.1.4 Java Data Base Connectivity (JDBC)

The JDBC API has been factored into two complementary components. The first component is API that is core to the java platform (the core JDBC 2.1 API) and comprises the updated contents of the java.sql package. This document contains the specification for the core JDBC 2.1 API . The second component, termed The JDBC 2.0 Optional packages API is described in a separate document. The java.sql package contains all of the additions that have been made to the existing interfaces and classes, in addition to a few new classes and interfaces. The new Javax.sql package has been introduced to contain the parts of the JDBC API which are closely related to other pieces of the java platform that the themselves optional pack-ages. Such as the java Naming and Directory interface (JNDI), and the java Transaction service (JTS).

In addition, some advanced features that are easily separable from the core JDBC API, such as connection pooling and row sets, have also been added to javax.sql. Putting these advanced facilities into an optional package instead of into core will help keep the core JDBC API small and focused. Since optional packages are downloadable, it will always be possible to deploy an application which uses the features in the JDBC optional package that will “run any-where”, since if an optional package isn’t installed on a client machine, it can be downloaded along with the application that uses it.

Over view of Features:

Result set enhancements:

The JDBC 1.0 API provided result sets that had the ability to scroll in a forward directionally. Scrollable result sets allow for more flexibility in the processing of results by providing both forward and backward movement through their contents. In addition, scrollable result sets allow for relative and absolute positioning. For example, it is possible to move to the fourth row in a scrollable result set directly, or to move directly to the third row following the current row, provided the row exists. The JDBC API allows result sets to be directly updatable, as well.

Batch updates:

The batch update feature allows an application to submit multiple update statement (insert/update/delete) in a single request to the database. This can provide a dramatic increase in performance when a large number of update statements need to be executed.

Row sets:

As its name implies, a row sets encapsulates a set of rows. A row set may or may not maintain an open database connection. When a row set is disconnected’ form its data source, updates performed on the row set are propagated to the underlying database using and optimistic concurrency control algorithm.

Row sets add support to the JDBC API for the java beans component model. A row set object is a bean. A row set implementation may be serializable.

JDBC driver and JDBC technology URL:

Connection pooling:

The JDBC API contains ‘hooks’ that allow connection pooling to be implemented on top of the JDBC driver layer. This allows for a single connection cache that spans the different JDBC drivers that may be in use. Since creating and destroying database connections is expensive, connection pooling is important for achieving good performance, especially for server applications.

Distributed transaction support:

Support for distributed transactions has been added as an extension to the JDBCE API. This feature allows a JDBC driver to support the standard 2- phase commit protocol used by the java Transaction service (JTS) API.

Other new features:

Support for character streams has been added. This means that character data can be retrieved and sent to the database as a stream of internationalized Unicode characters.

Methods to allow java.math . Big decimal values to be returned with full precision have also been added. Support for time zones has been added.

8.1.5 Introduction to Servlets:

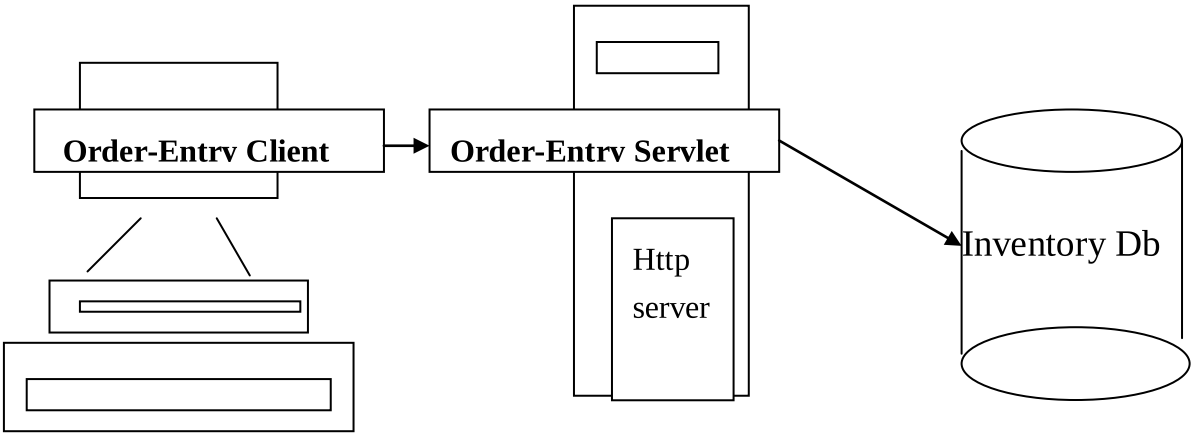
Servlets provide a java ™ based solution used to address the problems currently associated with doing server-side programming, Including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a java-based server. Servlets are to the server-side what applets are to the client-side—object byte codes that can be dynamically loaded off the net. They differ form applets in that they are faceless objects (with out graphics or a GUI component). They serve as platform the server side that can be used to dynamically extend server –side functionality.

A Servlet:

Servlets are modules that extend request/response-oriented servers, such as java-enabled web servers. 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. Unlike applets, however, servlets have no graphical user interface.



Servlets can be embedded in many different servers because the servlet API. Which you use to write servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the servlet API.

Uses of servlets:

Here are a few more of the many applications for servlets:

- Allowing collaboration between people. A servlet can handle multiple requests concurrently, and can synchronize requests. This allows servlets to support systems such as on-line conferencing.
- Forwarding requests. Servlets can forward requests to other servers and servlets. Thus servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries.

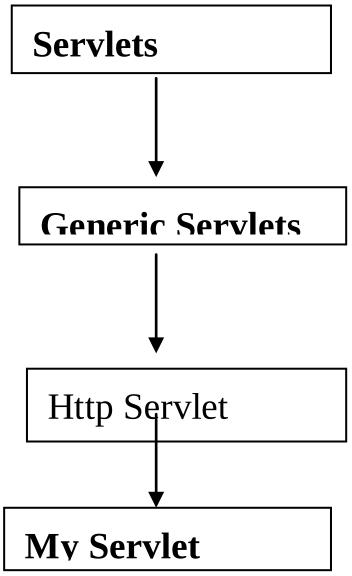
Architecture of the servlet package

The javax.servlet package provides interfaces and classes for writing servlets. The architecture of the package is described below.

The Servlet interface

The central abstraction in the Servlet API is the servlet interface. All servelets implements this interface, either directly or, more commonly, by extending a class that implements it such as HTTP Servlet.

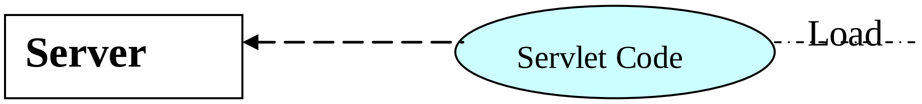
The servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

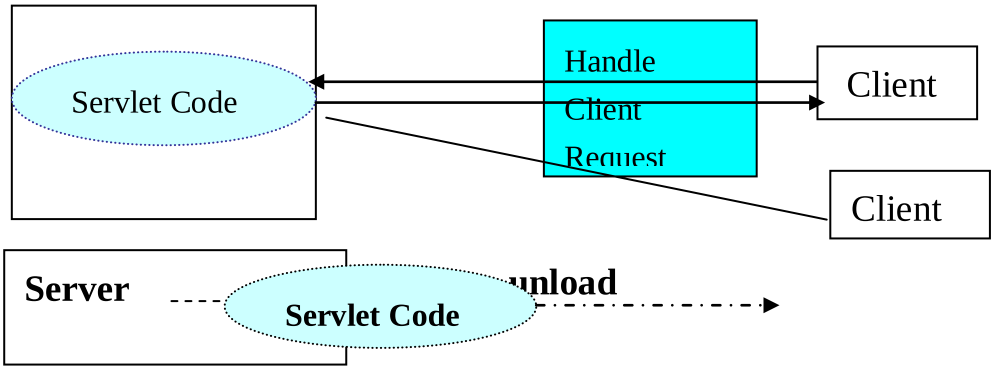


Servlet Lifecycle:

Each servlet has the same life cycle:

- ✓ A servlet loads and initializes the servlet
- ✓ The servlet handles zero or more client requests
- ✓ The server removes the servlet





Initializing a servlet:

When a server loads a servlet, the server runs the servlet's init method .initialization completes before client requests are handled and before the servlet is destroyed.

Even though most servlets are run in multi-threaded servers, servlets have no concurrency issues during servlet initialization. The server calls the init method once, when the server loads the servlet, and will not call the init method again unless the server is reloading the servlet. The server cannot reload a servlet until the server has been destroyed the servlet by calling the destroyed method.

The init method:

The init method provided by the HTTP Servlet class initializes the servlet and logs the initialization. To do initialization specific to your servlet, override the init() method following these rulers:

If an initialization error occurs that renders the servlet incapable of handling client requests, Throw and unavailable exception.

An example of this type of error is the inability to establish a required network connection.

- Do not call the system .exit method

Initialization parameters:

The second version of the init method calls the get init parameter method. This method takes the parameter name as an argument and returns a string representation of t he parameters value .if for some reason, you need to get the parameter names, use the get parameter Names method.

Destroying a servlet:

Servlet run until the servers are destroys them, for example at the request of a system administrator. When a server destroys a servlet, the server runs the servlet's destroy method. The method is run once. The server will not run that servlet again until after the server reloads and reinitializes the servlet. When the destroy method runs, another thread might be running a service request.

8.2 About Oracle 9i:

Oracle is a multi-user relational database management system .it is a software product that specializes in managing a single , shared set of information among many concurrent users, oracle products are based on a concept known as client/server technology. The concept involves segregation the processing of an application between two systems. One performs all activities related to the database (server) and the other performs the activities that help the user to interact the application (client).

A client or front-end database application also interacts with the database by requesting and receiving information from the database server. It acts and interface between the user and the database.

The database server and back-end is used to manage the database tables optimally among multiple clients who concurrently request the server for the same data. It also enforces data integrity across all client applications and controls database access and other security requirements.

Any programming environment used to create containers, to manage human data, in the conceptualization as a data management system. Traditionally, the block of human data being managed is called a database. Hence in very simple terms, these programming environments can the conceptualized as database management systems.

All databases management systems allow users to create containers for data storage and management. These containers are called 'cells'. The minimum information that has to be given to oracle for a suitable container to be constructed, which can hold free from human data, is

The cell name
The cell length
A cell can also be called as a field

8.2.1 Oracle9i Database Features:

Important features are introduced with the release of the oracle9i database .Including advancements in Real Application Cluster, systems management, availability scalability, and security, the features are.

Availability:

- Enhanced Disaster Recovery Environment

Many tasks for managing a stand by database are now automated. Log Miner now supports log queries based on changes and has comprehensive log analysis for additional data types. Plus and easy-to-use graphical user interface.

- Precision Database Repair:

Oracle9i includes better prevention and improved handling of disk corruption, and is able to restore the database to a consistent state after recovery.

Scalability:

- Scalable Session State Management :

The footprint required for each user on Oracle9i is substantially reduced, allowing more users to be hosted on the same, or larger, hardware platforms.

- Fine-Grained, Automatic Resource Management :

Administrators gain more granular control over resources through new features, and can specify the maximum active sessions per consumer group.

Security:

- Strong Three-Tier Security:

  Three-tier security is enhanced by proxy authentication, including credential proxy of x.509 certificates or distinguished name (DN). Support for thick JDBC, connection pooling for application users (thick and thin JDBC and OCI), and integration with LDAP.

- Improved User and Security Policy Management :

  A large number of enhancements have been made in a areas ranging from management of network naming and password –based user management to new replication queue management and reconciliation tools.

- Data Encryption and Label Security :

  Oracle Label security provides and out-of –the –box fine-grained access control solution. Encryption enhancements support most popular algorithms for encryption and data integrity.

8.2.2 Basic Database Concepts:

A database is a corporate collection of data with some inherent meaning. Designed, built and populated with data for a specific purpose. A database stores data that is useful to us. This data is only a part of the entire data available in the world around us.

To be able to successfully design and maintain databases we have to do the following.

- ✓ Identify which part of the world's data is of interest to us.
- ✓ Identify what specific objects in that part of the world's data are of interest to us.
- ✓ Identify a relationship between the objects.
- ✓ Hence the objects, their attributes and the relationship between them that are of interest to us are still owed in the database that is designed , built and populated with data for a specific purpose.

CHARACTRISTICS OF A DATABASE MANAGEMENT SYSTEM:

- It represents a complex relationship between data.
- Keeps a tight control of debtor redundancy.
- Enforces user-defined rules to ensure integrity of table data.
- Has a centralized data dictionary for the storage of information pertaining to data and its manipulation.
- Ensure that data can be shared across applications.
- Enforces data access authorization has automatic, intelligent backup and recovery procedures for data have different interfaces via which users can manipulation data.

RELATIONAL DATABASE MANAGEMENT:

A relational database management system uses only its relational capabilities to manage the information stored in its databases.

INFORMATION REPRESENTATION:

All information stored in a relational database is represented only by data item values, which are stored in the tables that make up the database. Associations between data items are not logically represented in any other way, such as the use of pointers form one table to the other.

LOGICAL ACCESSIBILITY:

Every data item value stored in relational database is accessible by stating the nature of the table it is stored in , the name of the column under which it is stored and the value of the primary key that defines that row in which it is stored.

INTEGRITY CONSTRAINTS:

Constraints that apply to entity integrity and referential integrity are specifiable by the data language implemented by the database management system and not by the statements coded into the application program.

DATA LANGUAGE:

The relational database management system may support many types of languages for describing data and accessing the database .However, there must be at least one language that uses ordinary character strings to support the definition of data, the definition of views, the manipulation of data, constraints on the data integrity , information concerning authorization and the boundaries for recovery of units.

8.2.3 Introduction to SQL:

SQL is the language that all users must use to access data with in oracle database. SQL was developed in a prototype relational database management , systems R, by IBM in the mid of 1970's system R was described by Dr.E.F.CODD in November 1976 journal of R&D. in 1979 oracle corporation introduced the first commercial implementation of SQL. American National Standard Institute (ANSI) adapted SQL as the standard language for relational database management systems in October 1986.

FEATURES OF SQL:

- SQL is a non-procedural language i.e., you specify what information you require, not to get it.
- SQL is a language for all users.
- SQL is a common language for all relational database.
- SQL is a data administration language that defines that structure of the data base, controls the users to accept the data.
- SQL is a client/server language that allows application programs on Pc's connected via LAN to communicate with the database servers that stores shared data.

SQL is used for storing and retrieving information in Oracle. A table is a primary database object of SQL that is used to store data.

TYPE OF SQL COMMANDS:

SQL commands are divided into the following categories:

- Data Definition Language.
- Data Manipulation Language.
- Transaction control Language.
- Session control Language.

Data Definition Language:

DDL statements define and maintain objects and drop objects when they are no longer needed. Example of this type of commands are CREATE, ALTER, and DROP COMMANDS.

Data Manipulation Language:

DML commands are the most frequently used SQL commands. This are used to query and manipulates existing objects like tables, examples of these commands are INSERT, SELECT, UPDATE, and DELETE.

Transaction Control Language:

A transaction is a logical unit of work. All changes made to the database can be referred to as a transaction. Transaction control statements manage the changes made by DML statements .these statements are used for controlling data stored in the database. Example these types of commands are COMMIT, ROLLBACK, SAVEPOINT etc.

Data Control Language:

DCL provides users with privilege commands. The owner of database like tables has the sole authority over them. The owner can allow other database users access to the objects as per his/her discretion, example of these commands are GRANT, REVOKE.

Session Control Language:

SCL allows a user to control the properties of his current session including enabling and disabling roles and changing language settings . example of these commands are ALTERSESSION, SETROLE.

CHAPTER 9

CONCLUSION

DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION can be used for fresher and experienced candidates. Students can apply for the resume and to get the details and to print the resumes in proper format.

In the DEVELOPMENT OF A FUTURE RICH RESUME BUILDER APPLICATION, updation is not possible. Because of this software is not supported. So this can be done by further extended technology.

CHAPTER 10

BIBLIOGRAPHY

1. Bruce Eckel
   THINKING IN JAVA 2nd Edition
   Prentice Hall.

2. Date c j, 2000
   AN INTRODUCTION TO DATABASE SYSTEM, 7th Edition
   Adidison-Wesley, Person Education Asia Pvt. Ltd.

3. Herbert Schildt.
   JAVA 2 COMPUTER REFERENCE
   Mc Graw- Hill international Edition, United States.

4. Pressman R S, 2001
   SOFTWARE ENGINEERING: A PRACTITIONER’S APPROACH
   Mc Graw-Hill international Edition, United States.

5. Silberscharz, Korth , Sudarshan ,2002
   DATABASE SYSTEMS CONCEPTS, 4th Edition. Mc Graw-Hill intenational Edition , United States.