A Project Report on

**Resume Craft**

Submitted in partial fulfilment of award of

**BACHELOR OF TECHNOLOGY**

Degree In

Computer Science & Engineering

Submitted By

**Abhishek Rajpoot**

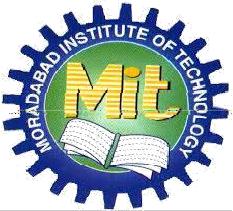
2100820100011

Under the guidance of

Mr. Ajeet Singh

Dr. Sachin K Singh

(Assistant Professor CSE Department)



**Department of Computer Science and Engineering Moradabad Institute of Technology, Moradabad (U.P.)**

Session: 2024-2025

**CERTIFICATE**

****

**TABLE OF CONTENT**

11-21

8.1 Economic Feasibility.................................................................................................

8.2 Technical Feasibility..................................................................................................

25-26

22-24

1. INTRODUCTION....................................................................................................................................

30

13. CONCLUSION.................................................................................................................

31

14. BIBLOGRAPHY ..............................................................................................................

12. SOFTWARE TESTING..................................................................................................

9. SYSTEM REQUIREMENT ANALYSIS..........................................................................

9

10

8.3 Operational Feasibility..............................................................................................

10

8

6.5 Limitations of Proposed System................................................................................

8

6.4 Advantages of Proposed System................................................................................

8

6.3 Proposed System........................................................................................................

8

6.2 Drawbacks of the Existing System............................................................................

7

1. PROBLEM DEFINATION...............................................................................................

7

5.2 Project Overview......................................................................................................

6

5.1 Introduction...............................................................................................................

11. SOFTWARE IMPLEMENTATION................................................................................

10. SYSTEM DESIGN..........................................................................................................

9

8. FEASIBILITY STUDY.......................................................................................................

8

7.SYSTEM REQUIREMENT SPECIFICATION..................................................................

8

6.1 Existing System.........................................................................................................

6

5

1. ACKNOWLEDGEMNT..............................................................................................................................................

4

1. ABSTRACT......................................................................................................................................

2

1. CERTIFICATE..................................................................................................................................

1

1. TITLE PAGE ....................................................................................................................

27-29

**ABSTRACT**

The Project “**Resume Craft**” 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.

By eliminating manual errors and offering a seamless experience, this system enhances productivity and efficiency. Its database-driven approach ensures data integrity, easy retrieval, and scalability, catering to a wide range of organizational needs. In essence, the “AI Resume Builder” serves as a dynamic and essential tool for bridging the gap between job seekers and employers, fostering professional growth and organizational success.

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to my teacher, Mr. Ajeet Singh, Dr. Sachin K. Singh for their invaluable guidance and support throughout the development of Resume Craft project. Their insights and encouragement were instrumental in helping me expand my understanding of web development and overcome challenges along the way. I am also deeply grateful to our Director, Dr. Rohit Garg, for providing me with this opportunity, allowing me to explore new technologies and conduct in-depth research that enriched my skills and knowledge.

Additionally, I would like to extend my heartfelt thanks to my parents for their unwavering support and encouragement, which motivated me immensely during this journey. Their understanding and patience played a crucial role in the successful completion of AI Resume Builder, and I am genuinely appreciative of their constant support.

**Abhishek Rajpoot**

**2100820100011**

**INTRODUCTION**

**1.1** **Introduction**

The Project title as “**Resume Craft”** 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 **“Resume Craft”** targeted for Noble Organizations is developed for the benefit of the users forproviding 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 them skills.

**PROBLEM DEFINITION**

The Project “**Resume Craft**” 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

: 80GB

**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

* Economic feasibility
* Technical feasibility
* Operational feasibility

**3.1 Economical Feasibility**

Economic 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 **“Resume Craft”** needsthe 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 inhandling 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.

**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 modelling techniques used in the system analysis avoids ambiguity by using precise modelling 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 sent 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. Modelling 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.

**Key Characteristics of the RAD Model**

1. **Incremental Development**

* RAD involves breaking down the entire project into smaller modules or components, which are developed independently in parallel.
* Each module is completed, tested, and integrated to form a fully operational system.

1. **User Involvement**

* Users play a critical role in the development process, providing continuous feedback during the iterative cycles.
* This ensures that the end product meets user expectations and is highly functional.

1. **Component-Based Approach**

* The use of pre-existing components or modules significantly reduces development time and effort.
* Developers focus on assembling these components rather than building everything from scratch.

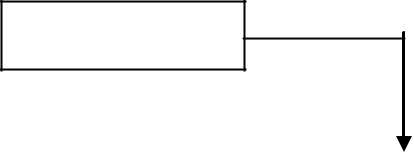
The RAD model strikes a balance between speed and quality, making it a preferred choice for projects with well-defined requirements and tight deadlines. By emphasizing iterative development, user involvement, and component reusability, the RAD model not only accelerates the development process but also ensures high user satisfaction. However, its success heavily depends on resource availability, requirement stability, and the team's expertise. When applied correctly, RAD can lead to the rapid creation of functional, high-quality systems that meet user needs effectively.

Diagram:

**Communication**

**Planning**

**Modeling**



Team #n

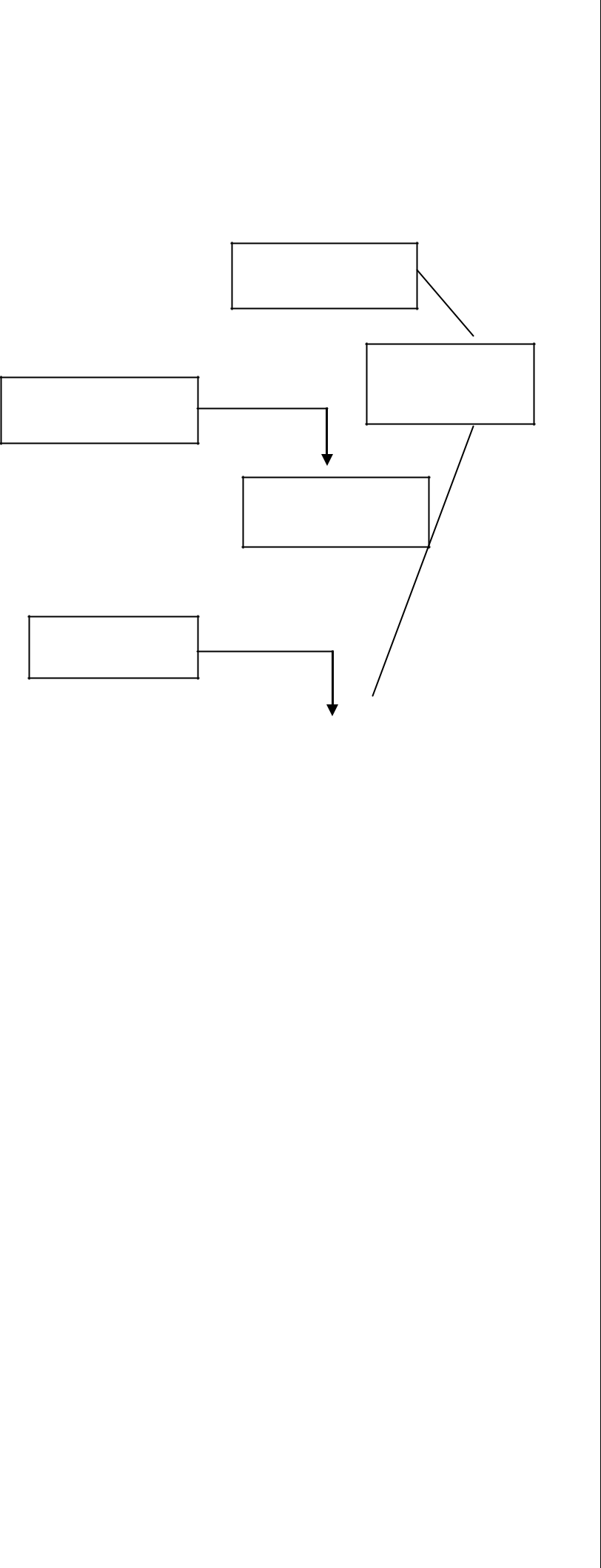
**Modeling**

Team #2

**Modeling**

Team #1

**Construction**



**Deployment**

**Construction**

**Construction**

**4.2 Entity-Relationship Diagram (ERD)**

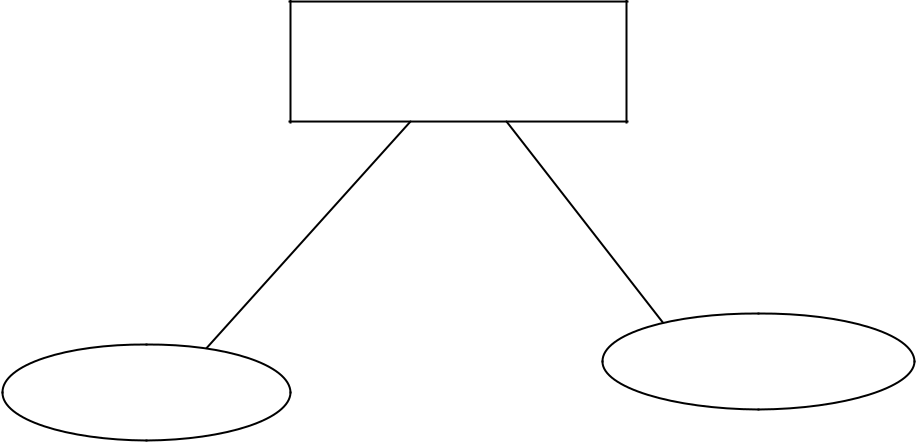
The Entity-Relationship Diagram depicts a relationship between data objects. The ERD is the notation that is used to conduct the data modelling activity. The attributes of each data object noted in the ERD can be described using a data object description.

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 labelled rectangles. Relationships are indicated with labelled 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 modelling 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:

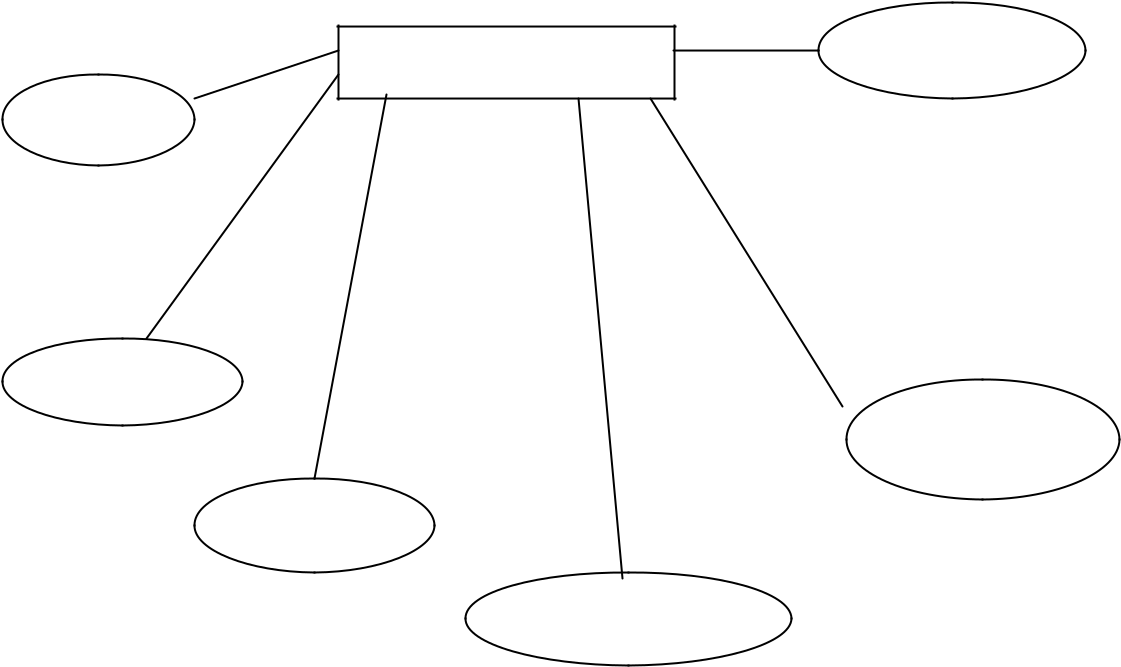


ADMIN

PASSWORD

USER ID

Registration:



REG

Regno

Fname

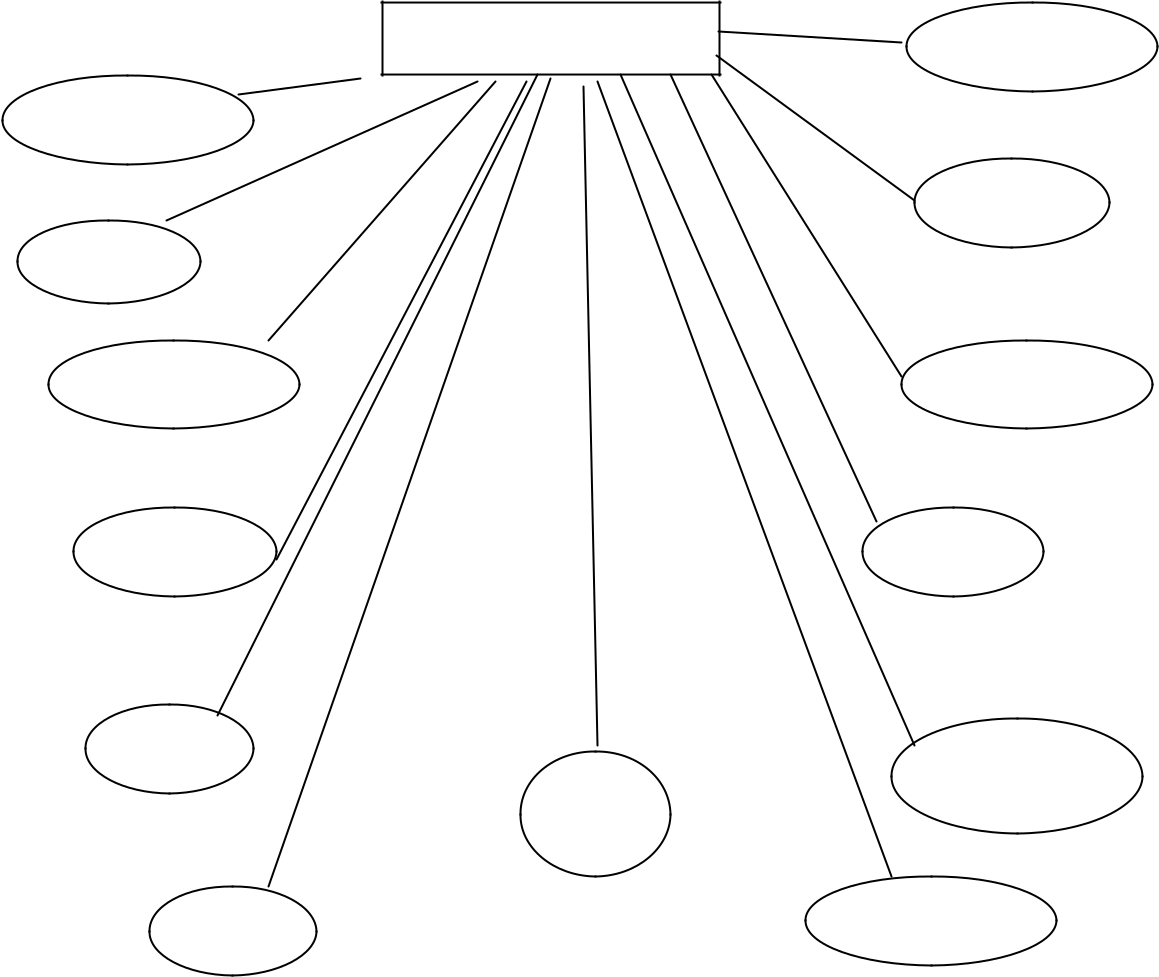
Lname

Sex

Dob

Eid

Personal Details:



Personal Details

Regno

Name

Fname

Addr

Sex

obj

Eid

hobbies

Lang

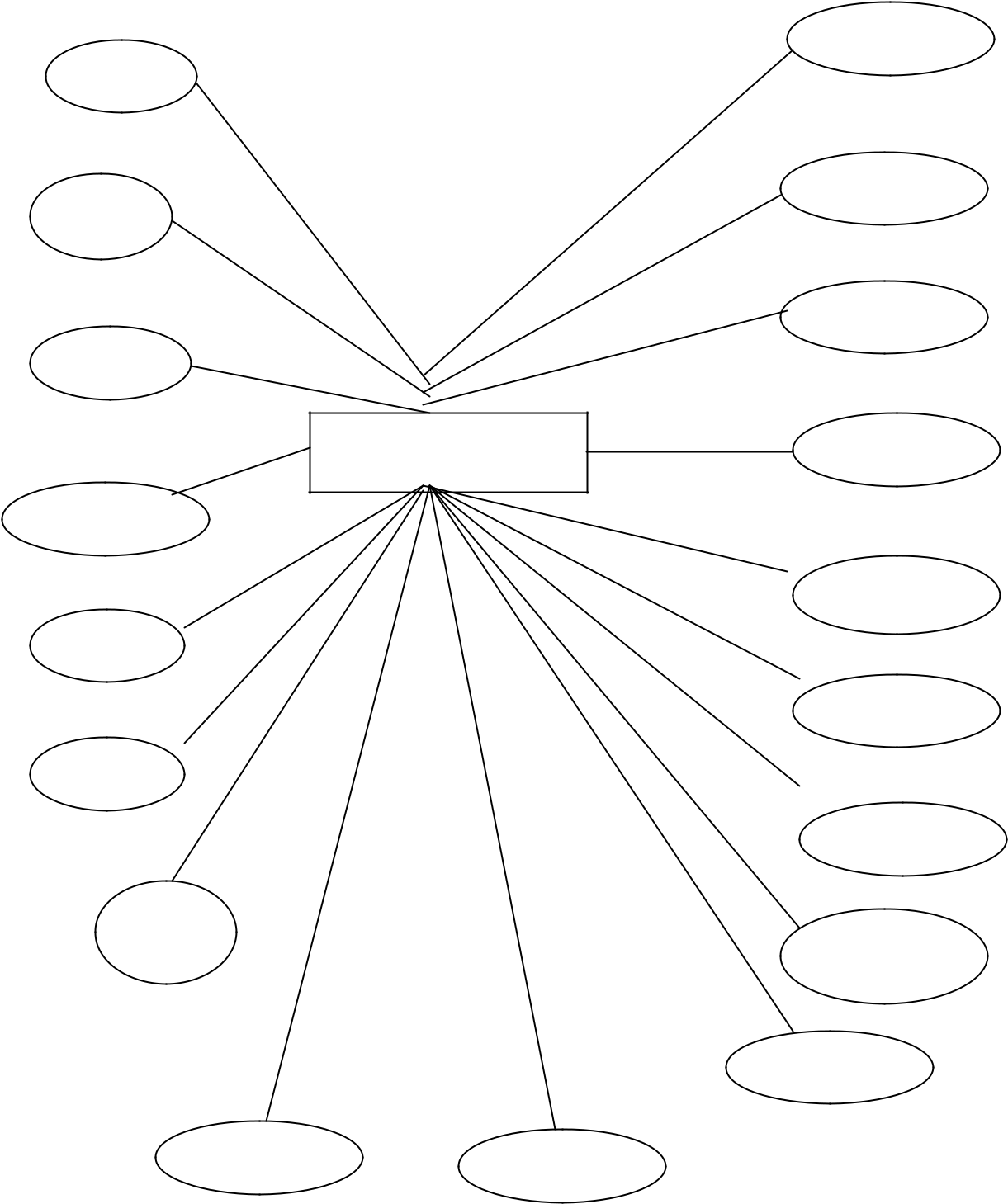
Nationality

Dob

religion

contno

Educational details:



Rid

Mq

Myr

Educational

Mper

Uq

Uu

Uyr

|  |  |  |
| --- | --- | --- |
| Ptitle | Pdes |  |
|  |  |

Tper

Tyr

Tu

Tq

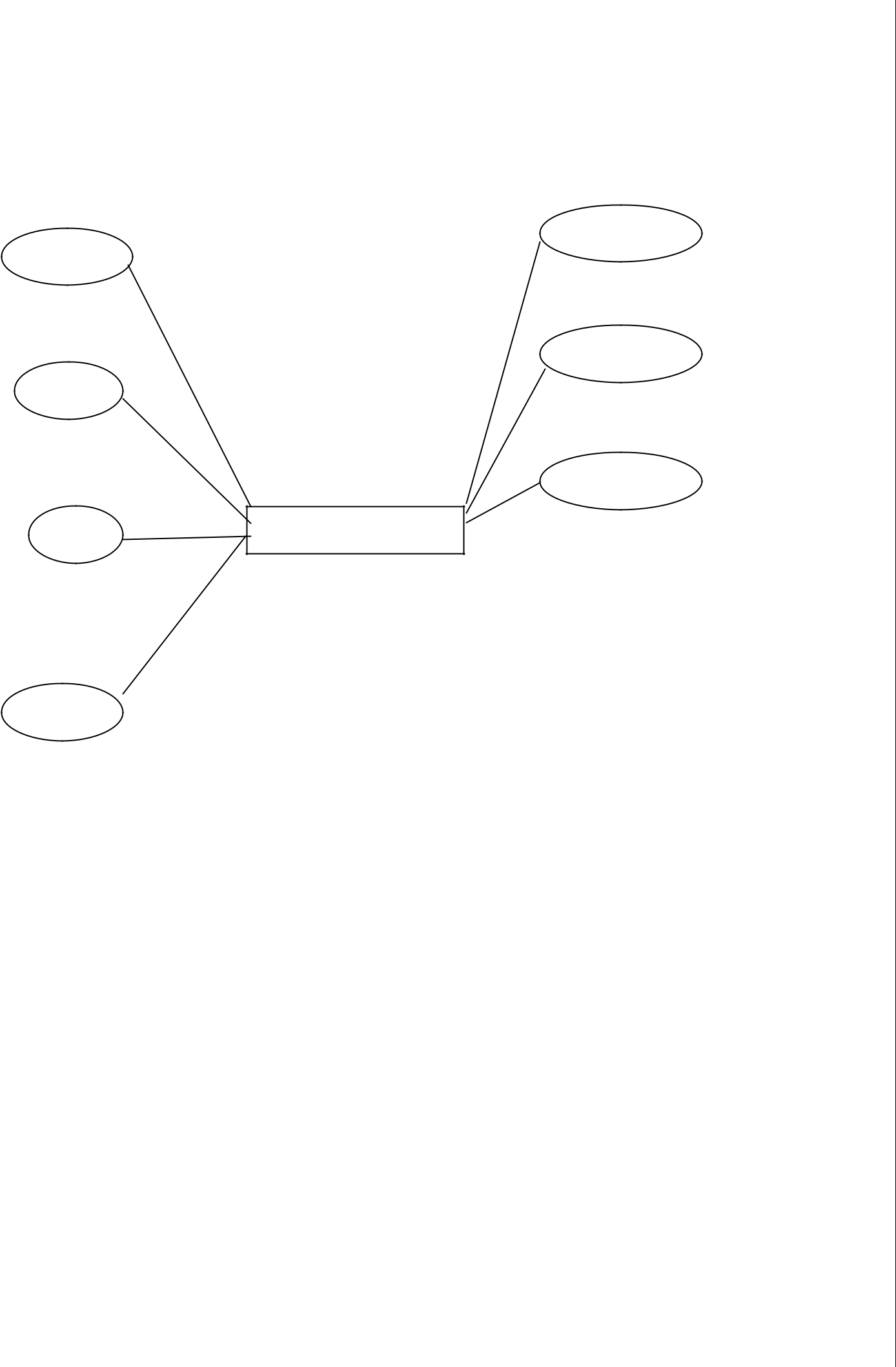
Iper

Iyr

Iu

Iq

Uper



Skills:

Regno

Pskils

Os

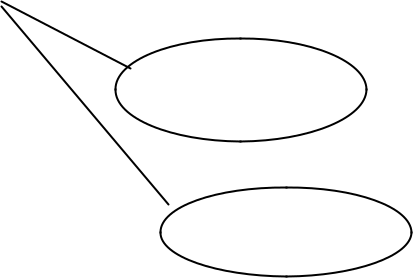
Plang

spl

Pack

Web

Skills



pdesc

Dbms

**4.3 Data Flow Diagram (DFD)**

The data flow diagram (DFD) is one the most important modelling 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.

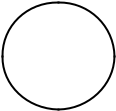
A level 1 DFD, also called as top-level DFD, represent the system with major modules and data stores. The other levels will show each module in the top-level DFD in a more detailed fashion.

***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.

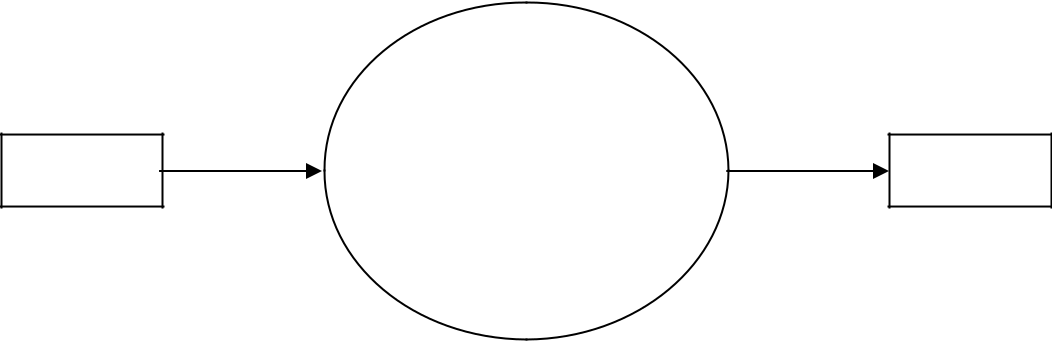


**Source or Destination of data:** External sources or destinations of data, which may be people or organizations or other entities.

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

User

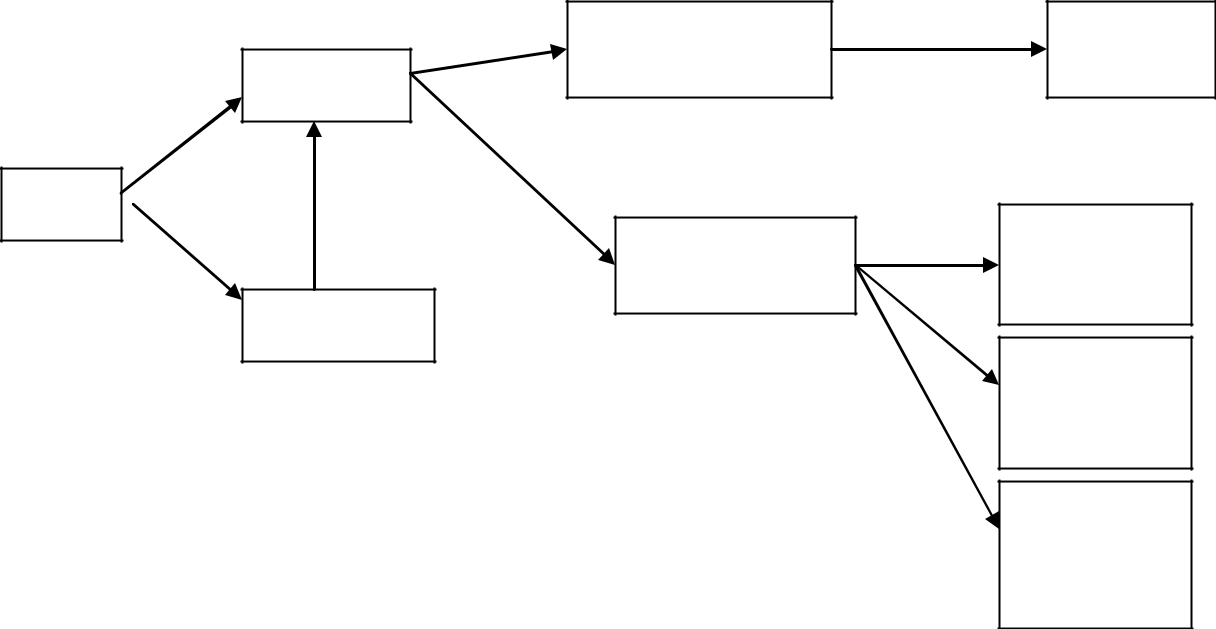
|  |  |  |  |
| --- | --- | --- | --- |
| Input | Resume Builder | Output |  |
| Application |  |
|  |  |  |



Resume

**Fig 4.2 Context-level DFD (Level 0)**

**RESUME BUILDER (Data Flow Diagram)**



**Login**

**User**

**Registratio**

**Build resume**

**Get resume**

**Resume**

**Resume**

**format1**

**Resume**

**format2**

**Resume**

**format 3**

**Operational-level DFD (Level 1)**

**4.4 Data Dictionary**

A data dictionary is a catalogue – a repository – of the elements in the system. As the name suggests, these elements centre 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 |  |  |  |
|  |  |  |  |  |  |  |  |

**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 behavioural 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.

**The Iterative Nature of Software Design**

Software design is inherently iterative, meaning it is revisited and refined throughout the development lifecycle. The requirements of the software are continuously translated into more concrete designs. These designs are created at a high level of abstraction, allowing for the identification of the primary components and their interactions. As the design process progresses, the abstraction level gradually decreases, moving towards a detailed design that can be directly translated into code.

**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 specifications of the modules can be system design are

the interconnection of the modules or how the satisfied is decided. Some properties for a software

* 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
* 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.

**SOFTWARE IMPLEMENTATION**

**Introduction**:

Implementation is the process of assuring that the information system is operational and then allowing users take over its 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

**Direct Conversion**

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

**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.

**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.

**Phased Conversion**

Phased conversion means introducing a system time, waiting until that one is operating properly before

in stages, one component or module at a introducing the next.

**Hardware Environment**

CPU : Pentium 4, 1.7 GHz

RAM : 512 MB

Hard disk : 80 GB

**Software Environment**

Operating Environment : Windows200/XP

Back-End : ORACLE

Front End : JSP

Browser : IE7

**SYSTEM TESTING**

**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.

**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.

**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.

**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.

**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.

**Verification and Validation:**

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 ages 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).

Verification is a systematic approach to check whether the product is being developed correctly. It ensures that the software adheres to the technical specifications and design documents established during the planning phase. The primary question it addresses is: *“Are we building the product, right?”* Verification focuses on the internal consistency and correctness of the software, ensuring compliance with the defined architecture, design patterns, and coding standards.

**Interrelation of Verification and Validation**

Verification and validation are interdependent and iterative processes. Verification typically precedes validation and lays the groundwork for it. For example, verifying a requirement specification ensures that the functional design aligns with user needs, which is then validated through functional testing. Similarly, code verification ensures that the implementation conforms to the design, which is later validated through code execution tests.

Verification and validation are integral to delivering a high-quality software product that meets user needs and adheres to technical specifications. While verification ensures the correctness of the development process, validation ensures the relevance and functionality of the end product. Together, they form a comprehensive quality assurance strategy that enhances user satisfaction, reduces risks, and ensures successful project outcomes.

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).

**CONCLUSION**

The development of a future-rich resume builder application using the MERN stack and AI-driven technologies offers an innovative solution for both fresher and experienced candidates. This platform provides users with an intuitive interface to build highly professional, customized resumes that align with industry standards. By leveraging AI for content optimization and generation, the application can analyze a candidate's input and suggest improvements, ensuring that the resume is well-crafted, relevant, and tailored for specific job roles.

The integration of real-time editing, customizable templates, and AI-powered suggestions allows users to quickly create and optimize their resumes for maximum impact. The platform's ability to provide keyword optimization for Applicant Tracking Systems (ATS) ensures that resumes are not only visually appealing but also optimized for digital recruitment processes.

While the current version of the application does not support automatic updates or real-time content generation, this limitation can be addressed with future advancements in technology. As AI and machine learning continue to evolve, the application could offer enhanced features such as real-time content generation, automated resume updates, and personalized career advice, further improving the user experience

.

In summary, a MERN-based, AI-driven resume builder application can significantly enhance the job application process, providing users with a powerful tool to create optimized, professional resumes that increase their chances of success in a competitive job market. As technology advances, this platform has the potential to expand its capabilities, offering even more advanced features that streamline the job search journey for candidates.

Moreover, the use of the MERN stack in building this resume builder application ensures a robust, scalable, and efficient platform capable of handling large numbers of users and data with ease. The integration of AI for content analysis and resume optimization not only enhances the quality of the resumes but also helps users create tailored documents that resonate with specific industries and job roles. The future evolution of this platform could include deeper personalization features, such as AI-driven career path suggestions, skill gap analysis, and integration with job portals for direct applications.

**BIBLOGRAPHY**

1. **González, G., & Salgado, M. (2019).** *AI-Powered Resume Optimization: A Case Study in the Job Application Process.* Journal of Artificial Intelligence Research, 35(2), 121-134.  
   This article discusses the role of AI in enhancing the job application process, particularly through the optimization of resumes. It highlights the impact of machine learning algorithms in improving content quality and relevance for specific job roles.
2. **Sweeney, L., & Tran, K. (2021).** *MERN Stack Development: Building Scalable and Dynamic Web Applications.* O'Reilly Media.  
   This book provides a comprehensive guide to using the MERN stack for building modern web applications. It covers the development process, from setting up the backend with MongoDB and Express to creating dynamic, user-friendly interfaces with React.
3. **Lee, H., & Kim, Y. (2020).** *The Role of Applicant Tracking Systems in Recruitment: A Study on Resume Screening Technologies.* International Journal of Human Resource Management, 32(4), 567-582.  
   The research explores how Applicant Tracking Systems (ATS) function in the recruitment process and the importance of optimizing resumes for ATS compatibility. It underscores the need for AI-powered tools that help candidates adjust their resumes to meet these systems' criteria.
4. **Patel, S., & Sharma, R. (2018).** *Machine Learning for Resume Generation and Optimization: Techniques and Best Practices.* International Journal of Computer Science and Engineering, 24(6), 2139-2152.  
   This article delves into how machine learning techniques can be applied to automatically generate and optimize resumes. It explores algorithms that can suggest improvements based on industry standards, keywords, and job descriptions, forming the backbone of AI-powered resume builder applications.
5. **Davis, B., & White, R. (2022).** *Designing User-Centred Applications: Best Practices in User Experience and Interface Design.* Springer.  
   This book discusses the principles of user-centred design and their application in creating intuitive, effective user interfaces. It is particularly useful for understanding how to create a seamless user experience in applications like resume builders, where simplicity and accessibility are key.
6. **Wang, L., & Zhang, Q. (2019).** *Resume Generation through Deep Learning and Natural Language Processing.* Journal of Artificial Intelligence in Education, 45(3), 401-418.  
   This research paper explores how deep learning and natural language processing (NLP) can be leveraged to automatically generate resumes based on minimal user input. It highlights how AI can enhance content accuracy and improve formatting, making resumes more competitive.
7. **Robinson, P. (2021).** *The Future of Job Applications: AI and Automation in Recruitment.* Career Development Quarterly, 69(1), 35-46.  
   This paper investigates the growing role of AI and automation in the recruitment process. It outlines how tools like resume builders can integrate AI to provide personalized, data-driven resume optimization to enhance candidates' chances of landing interviews.