**CAREER MANAGEMENT SYSTEM**

****

**Submitted by -**

**Harshit Sahu (20211414)**

**Kartekay Kaushik (20211421)**

1. **Gyanendra Singh (20211435)**

**Rahul Rai (20211450)**

**Submitted to -**

**Dr. Kamlesh Kumar Raghuvanshi**

**CERTIFICATE**

This is to certify that **Harshit Sahu**, **Kartekay Kaushik**, **N. Gyanendra Singh** & **Rahul Rai**, the students of BSc(H) Computer Science, Ramanujan College of Delhi University with Roll No. **20211414**, **20211421**, **20211435** & **20211450** respectively have made the project titled “Career Management System” in their individual capacities under my supervision for the fulfillment of Sem IV practical examination.

(Dr. Kamlesh Kumar Raghuvanshi)

**ACKNOWLEDGEMENT**

## We extend our profound gratitude to our project guide Dr. Kamlesh Kumar Raghuvanshi, for his interest, guidance and suggestions throughout the course of the project. We feel honored and privileged to work under him. He shared his vast pool of knowledge with us that helped us steer through all the difficulties with ease. This project would not have been possible without his guidance and we would like to thank him for his assistance.

Harshit Sahu

Kartekay Kaushik

N. Gyanendra Singh

Rahul Rai

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| Sl. No. | Topics | Page |
| 1. | Abstract | 5 |
| 2. | 1. Introduction 2. Purpose of Project 3. How It Benefits A User 4. Team Architecture | 6  7  8  9-10 |
| 3. | 1. Problem Statements 2. Project Overview | 11-12  13-14 |
| 4. | Requirement Analysis | 15-21 |
| 5. | Project Planning | 22 |
| 6. | Risk Management | 23-25 |
| 7. | Tier Architecture | 26-31 |
| 8. | System Design | 32-40 |
| 9. | Software Testing & Security | 41-43 |
| 10. | References | 44 |

**ABSTRACT**

The project “Career Management System” is built around the idea of providing students with the necessary guidance in selecting their desired career paths. It helps the students to track, develop and manage their career interests. The key component of this system revolves around the creation of a resource for the students to better aid them in being fully aware of all the available career options as per their skills, interests and department. It also provides the student with a step by step approach to advance towards their career choice. The “Career Management System” and each of its components allow for the ownership of the student’s career to be held and acted upon by the students themselves. The continual feedback allows for progressive improvement by the student and in-depth understanding of the steps required to be followed.

**INTRODUCTION**

A difficult phase for many students is when they come to the time of choosing a career for their future. In the process of making this decision, students are faced by many difficulties including lack of self-awareness and self-evaluation. If these problems are not resolved in time, then it creates anxiety in the student and makes them take inappropriate decisions in their career choice. This is where a “Career Management System” comes into play and aids the student in determining a suitable career path as per their skills, interests and field. Depending upon the credentials entered by the student, the system presents them with a wide array of career paths from which they can then find details about their desired jobs. The system nullifies the lack of self-awareness and self-evaluation in students.

**PURPOSE OF PROJECT**

“Career Management System” is a system using which students can receive a list of career options as well as details about them as per their credentials. It aims to eliminate the lack of self-awareness in students by providing guidance about possible jobs.

The purpose of this project is to provide a resource to the students using which they will become aware about the career path that they want to undertake.

This project is feasible because :-

1. The system contains minimal complex elements.
2. The system will have a basic GUI.
3. It can be operated by almost everyone.

**HOW IT BENEFITS A USER**

Choosing the right career can prove to be a dilemma for many students. Some are not aware about all of the available options & some are not aware at all. This lack of information creates confusion among the students & so they take unwanted decisions for their career. A “Career Management System” can prove to be a useful resource for the students in determining their desired jobs as per their interests. The system filters out all of the available career paths according to the details they enter like subjects, department & qualifications. From there, the students can further find details about their selected job including its requirements. This will completely eliminate the feeling of confusion present in students. It will also help them in deciding which career will suit them the most as per their interests.

**TEAM ARCHITECTURE**

There were 4 members involved in the making of this project which were then divided into 2 teams.

**TEAM 1 -**

**REQUIREMENT GATHERING & PLANNING**

**Requirement gathering** - If the requirements of the user are clear, then it is easy for the developer to fulfill their needs. They look into all the required resources and through proper planning with optimal cost estimation, they achieve the required software.

**Planning** - Planning identifies all deliverable services, facilities and also defines the working of the project. It starts with a specification of user demand that is to be met by the production. Then, choices are made as to which resources to include as they are limited. Optimally, only the most critical resources are used.

**TEAM 2 -**

**DESIGNING AND BACKEND**

**Designing** - It is the most important entity during software development. Without a proper design, it is very difficult to develop an appropriate software that fulfills nearly all of the user demands. Therefore, managing the designing phase is very crucial. A designer needs to ensure that their design can be easily understood by all the members of the developer team. Also, a proper design allows the coder to implement the planned system properly.

**Backend** - This section involves the use of databases and so it plays a key role in all of the systems. Therefore, the team working on this field must have full knowledge and experience to properly handle as well as manage all of the databases and their tools.

**PROBLEM STATEMENTS**

Nowadays, many students are confused about their future life. They don’t know how to tackle this situation. Rather than focusing on their career, they get stuck in between their friend’s & family’s opinions. So, to rectify this problem/situation we have developed a “Career Management System”.

In today's fast-paced and competitive job market, it is essential for individuals to develop their careers and remain relevant in their industries. However, many individuals struggle with identifying and pursuing the right career paths, developing the necessary skills and knowledge, and accessing the right resources and opportunities to advance their careers. This is especially true and challenging for individuals who are just starting out in their careers or seeking to make a transition to a new field.

To address this problem, a carrier development software can be developed that provides users with personalized guidance, resources, and tools to help them identify their career goals, develop the necessary skills and knowledge, and access the right opportunities to advance their careers. The software should be user-friendly and accessible to individuals of all levels of experience and expertise. It should leverage data analytic and artificial intelligence to provide personalized recommendations based on users' interests, skills, and career goals. The software should also include features such as career assessments, job matching, skill development resources, and career coaching services to support users in their career development journeys. By developing such a carrier development software, individuals can access the support and resources they need to navigate the rapidly changing job market, to succeed in their respective career paths.

**PROJECT OVERVIEW**

**Project Overview** - The carrier development software project aims to create a user-friendly and personalized software platform that helps individuals navigate the job market and develop their careers. The software will include features such as career assessments, job matching, skill development resources, and career coaching services. The project will leverage data analytic and A.I to provide personalized recommendations to the user based on their data such as interests, skills, and career goals.

**Target Users** - The carrier development software is intended for individuals of all levels of experience and expertise who are seeking to advance their careers. The software will be particularly useful for individuals who are just starting their careers or seeking to make a transition to a new field.

**FEATURES**

1. **Career Assessments** - The software will provide users with various career assessments to help them identify their strengths, interests, and career goals. These assessments may include personality tests, skills assessments, and career interest inventories.
2. **Job Matching** - The software will use data analytic to match users with job opportunities that align with their skills, interests, and career goals. This will help users find job openings that are a good fit for them and save them time in the job search process.
3. **Skill Development Resources** - The software will provide users with access to various skill development resources, such as online courses, tutorials, and workshops. These resources will help users develop the skills and knowledge they need to advance in their careers to find better opportunities.

**REQUIREMENTS**

1. **SPECIFIC REQUIREMENTS**

**External Requirements** - The external system is to take full responsibility for storage functions as well as database management and database control for the entire database. The interfaces in this section are specified by documenting : the name and description of each field, source/input, destination/output, ranges, accuracy, timing, display formats, organization and data formats. The user interface required to be developed for the system should be user-friendly and attractive. The interface between the user and the system will be WIMP (Windows, Icons, Menu, Pointers) keeping in mind that the system is to be run in a web browser. All operations will be off point and click based with all the actions performed through buttons and menu.

**Buttons** - The button is activated when the user clicks on the left click of the mouse within the bounds of the button. And thus the action associated with it will be carried out.

**Menu** - All operations will be carried out as needed.

**Hardware Requirements** :

• 20 GB HDD, 256 MB RAM

• Pentium IV Processor

• Input Devices : Keyboard, Mouse

• Output Devices : Monitor, Printer

**Software Requirements** :

• Operating System : Windows XP or above

• Browser : Any web page supporting browser

• Software : SQL(RDBMS), VS Code(Code Editor)

1. **FUNCTINAL REQUIREMENTS**

• Create an account.

• Log into the system.

• Navigate the system's menu.

• Select a field from the menu.

• Customize options for the selected field.

• Add/Remove a field to their current plan.

• Review the current plan.

• Provide payment details for premium services.

**Menu Management System -** The Menu Management System will be available only to developers and will, as the name suggests, allow them to manage the menu that is displayed to users of the software. The functions provided by the Menu Management System will allow the user to perform these actions using Graphical Interface :-

• Add/update/delete category to/from the menu.

• Add/update/delete item to/from the menu.

• Add/update/delete option for a given field.

• Update default options for a given field.

• Update additional required information.

**Retrieval System -** Like the Menu Management System, it is designed to be used only by the developers & provides the following functionality :-

• Retrieve new registrations from the database.

• Display the requests/queries in a graphical way.

• Mark a request/query as having been processed and remove it from the list of active orders.

**User Interface Specifications -** Every component of the system will have their own individual interfaces. These are explained below :-

**Web Accessing System -** Users of the Web Accessing System will interact with the application through a series of simple interfaces. Each category has its own form associated with it which presents a drop down menu for choosing the required item. Adding an item to the field is accomplished by a single button click. Users select which category of career they would like to choose, and therefore which form should be displayed, by navigating a menu bar, an approach which should be familiar to most users. Entering subscription and payment deals is done in a similar manner. The user is presented with a form and must fill the required fields, which include both drops down and text boxes, before activating and receiving a confirmation text/mail.

1. **NON-FUNCTIONAL REQUIREMENTS**

**Performance Criteria** -

**Time** - The time required for the acceptance of a user request should be as minimum as possible.

**User-friendly** - The system should be user friendly. The interface should be kept simple and uncluttered. Since different types of people will utilize the system, it should be very easy for them to use & understand.

**Flexibility -** The system should be simple in nature for fast & easy modifications whenever it is required.

**Extensibility -** The system should have the capability to having new features or functions being added to it without modifying the underlying foundations.

**Portable -** The system should be portable on any kind of platform and be easily available on websites.

**Reusable -** The existing software systems should allow the development of new software systems. This will reduce the time and effort needed to create software systems from the ground up.

**DESIGN TECHNIQUES**

The design of this system has been done using the following technologies :-

• HTML

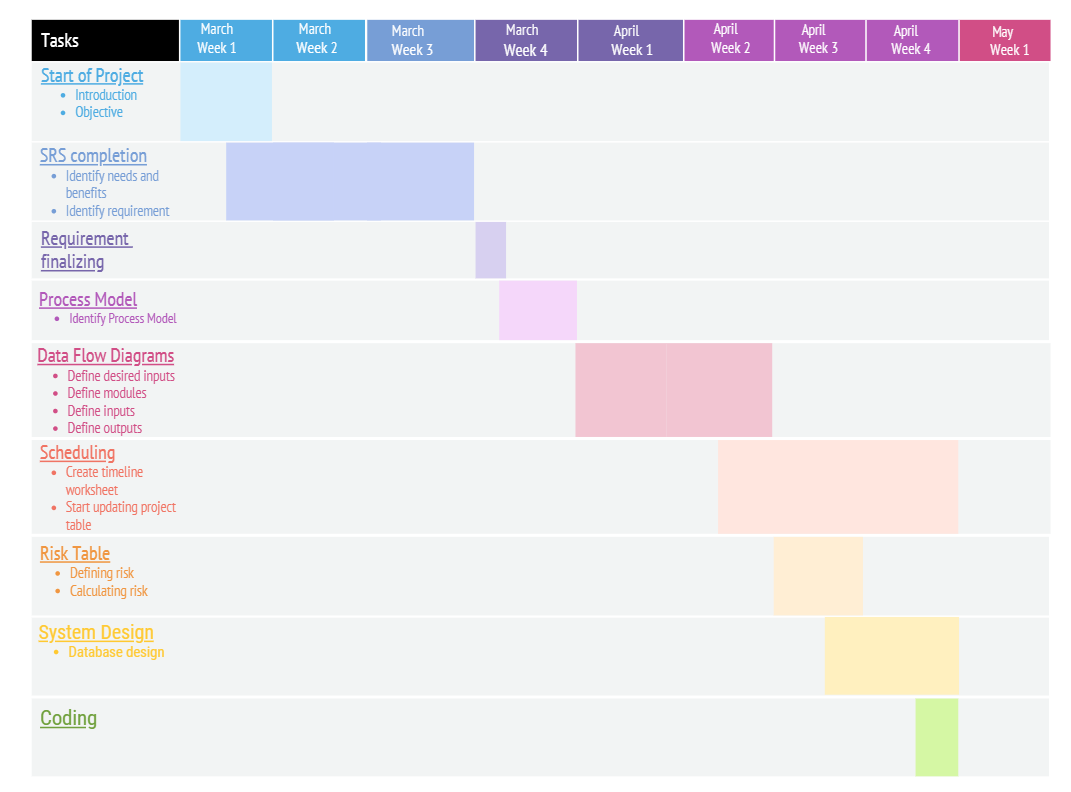
• CSS

• PHP

• MySQL

• JavaScript

**PROJECT PLANNING**

****

**RISK MANAGEMENT**

Identifying potential risks and developing a plan to mitigate, monitor and manage risks is of paramount importance. Risk analysis enables to build a risk table by providing detail guidelines in identification and analysis of risk.

Points to be considered are :-

* Risk Avoidance
* Risk Monitoring
* Risk Management and Contingency Plan

**Impact Values for Risk Table**

1. Catastrophic
2. Critical
3. Marginal
4. Negligible

By considering some external and internal factors here are some risks that may arise and further a risk management table is made :-

* Lack of clear product vision.
* No one may be inadequate to do the job.
* Delivery Date may extend.
* Customer may change the requirements.
* End User may resist the system (students mainly).
* Budget Problem.
* Lack of documentation.

**RISK MANAGEMENT TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Category** | **Probability** | **Impact** | **RMMM** |
| Lack of clear Product vision. | Project  Risk | 60% | 2 | * Ensure that Requirements are clearly understood. * Constantly monitoring and revising the requirements. |
| Delivery deadline may be tightened. | Business  Risk | 80% | 2 | * Schedule made should be realistic and achievable. * Monitor that efforts are according to schedule. |
| No. of people may be inadequate to do the job. | Project  Risk | 60% | 2 | * Organize a task network * Assign a backup staff. Member as a third party for testing and review. |
| Customers will change requirements. | Project  Risk | 40% | 2 | * Update the employer regularly about the status. * Get customer’s feedback periodically. |
| End user may resist the system. | Business  Risk | 40% | 1 | * Involve the end users in the development of the system. * Develop techniques to attract the end users. |
| Budget Problem | Business Risk | 40% | 2 | * Budget should be decided on the basis of past project experience. |

**TIER ARCHITECTURE**

The 3 tier architecture consists of three layers :-

**Presentation Layer** - The website or windows forms

application is called the presentation layer. The

presentation layer is the most important layer simply

because it's the one that everyone sees and uses. Even with a well-structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the system. The presentation layer is the form where we design using the controls like text box, labels, command buttons.

**Business Layer** - Though a website could talk to the

data access layer directly, it usually goes through another layer called the business layer. This layer is a class which we use to write the function which works as a mediator to transfer the data from Application or presentation layer data layer. In the three-tier architecture, we never let the data access

layer interact with the presentation layer. This layer is also a class where we declare the variable corresponding to the fields of the database which can be required for the application and make the properties so that we can get or set the data using these properties into the variables. These properties are public so that we can access its values. One of the best reasons for reusing logic is that applications that start off small usually grow in functionality. For instance, a company begins to develop a website, and as they realize their business needs, they later decide to add a smart client application and windows service to supplement the website. The business layer helps move logic to a central layer for "maximum reusability".

Business layer have two roles :-

• Client application

• Server component

**An example of Business Layer** - The Business layer has functions which take the parameters from the example given in the presentation layer As the user inputs the data values. corresponding functions are called in the business layer which is further passed on through the data layer where corresponding procedures are called and the data is updated.

**Data layer** - The key component to most applications is the data. The data has to be served to the presentation layer somehow. The data layer is a separate component whose sole purpose is to serve up the data from the database and return it to the caller. This layer is also a class which we use to get or set the data to the database back and forth. This layer only interacts with the database. We write the database queries or use stored procedures to access the data from the database or to perform any operation to the database.

**ADVANTAGE OF 3 TIER ARCHITECTURE**

• Client-Server architecture is 2-Tier architecture because the client does not distinguish between the Presentation layer and the business layer.

• The increasing demands on GUI controls caused

difficulty to manage the mixture of source code from GUI and Business Logic.

• Further, Client Server Architecture does not support Change Management. Let suppose that the government increases the Entertainment tax rate from 4% to 8%, then in the Client-Server case, we have to send an update to each client and they must update synchronously on a specific time otherwise, we may store invalid or wrong information.

• The Client-Server Architecture is also a burden to

network traffic and resources. Let us assume that

about five hundred clients are working on a data

server then we will have about five hundred ODBC

connections and several ruffian record sets, which

must be transported from the server to the clients. This categorization of the application makes the function more reusable easily and it becomes too easy to find the functions which have been written previously. If the programmer wants to further update in the application then its easy to understand the previously written code and make the update.

**DISADVANTAGES**

• Increase complexity effort

• More difficult to build 3 tier architecture.

• The points of communications are doubled. Maintenance tools are currently inadequate for maintaining server libraries.

**WHY INCREMENTAL MODEL?**

1. Incremental models are advisable where

requirements are clear and the development time is

less. The striking feature of the incremental model

is that each module can be completed and released

when the requirement rises because of lack of time.

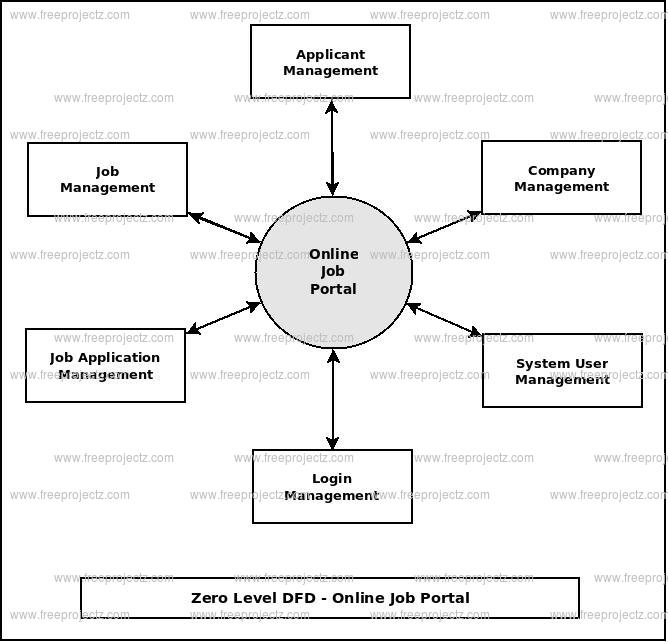
2. As in our system, many of the modules are not

interrelated so they can be released in isolation. The user can thus get a feel of these modules and give his feedback which can be utilized for making the software more user-friendly and in line with the user requirements.

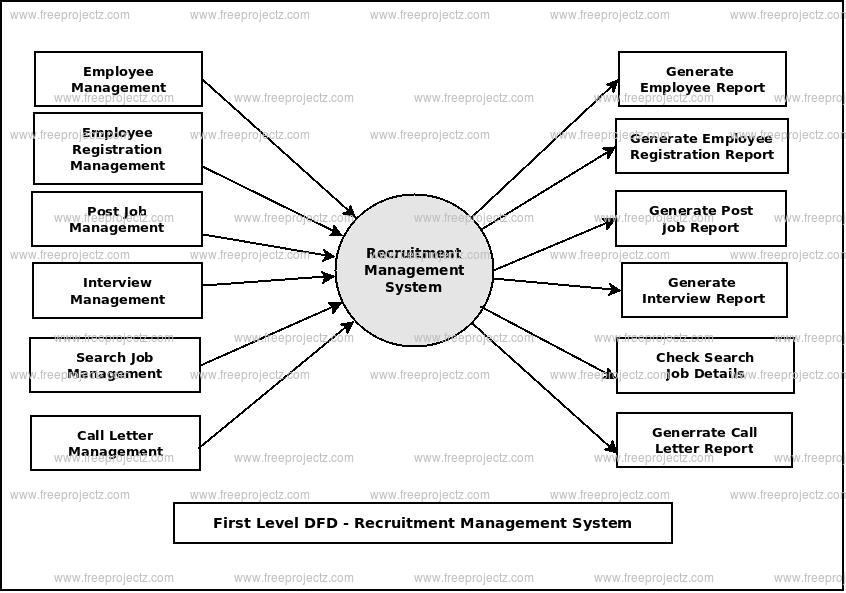
**SYSTEM DESIGN**

**DATA FLOW DIAGRAMS**

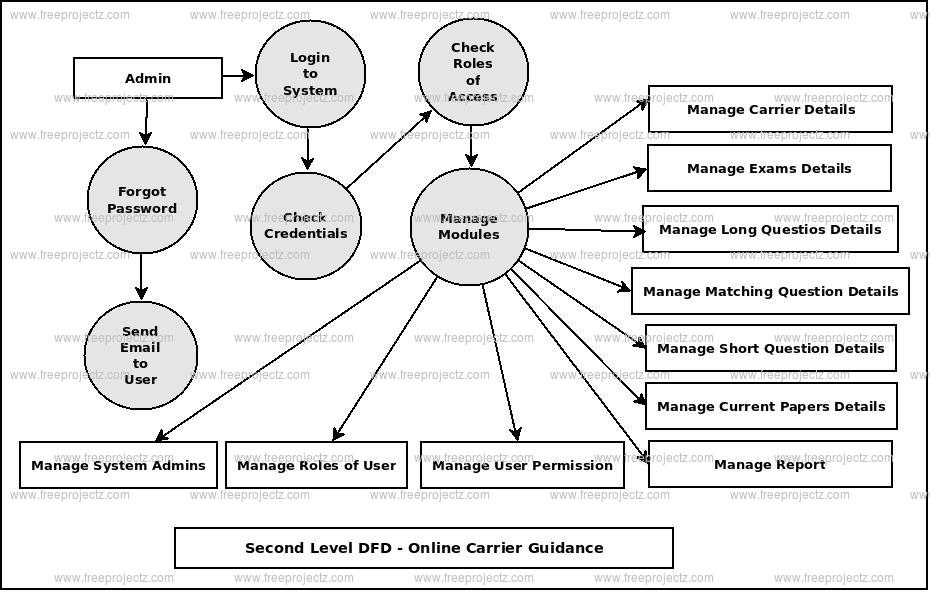
**Level 0 Diagram**



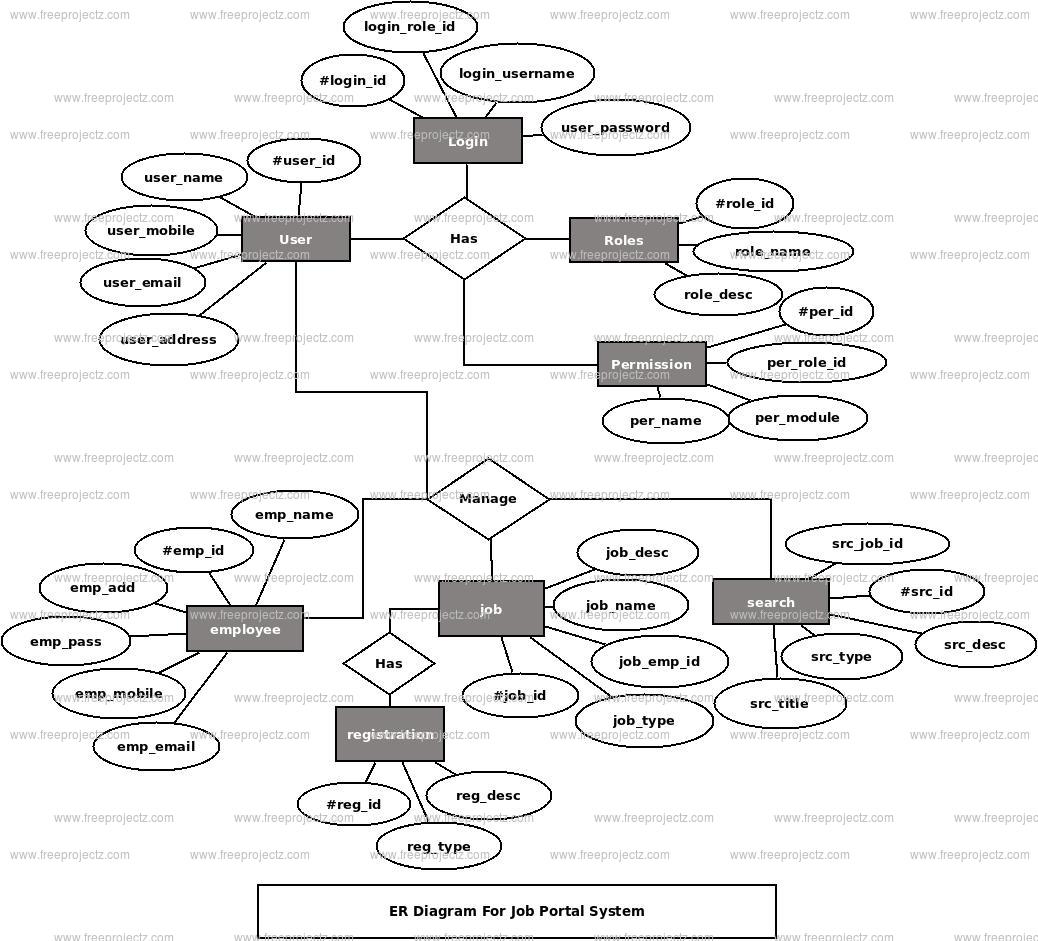
**Level 1 Diagram**



**Level 2 Diagram for Admin**



**ER DIAGRAM**



**DATABASE DESIGN**

USER TABLE

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| UserID | int | Primary Key, Auto Increment |
| FirstName | varchar(50) | Not Null |
| LastName | varchar(50) | Not Null |
| Email | varchar(100) | Not Null, Unique |
| Password | varchar(100) | Not Null |
| CareerGoals | text |  |
| Resume | varchar(255) |  |
| CoverLetter | varchar(255) |  |
| OtherMaterials | varchar(255) |  |

JOB TABLE

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| JobID | int | Primary Key, Auto Increment |
| Title | varchar(100) | Not Null |
| Company | varchar(100) | Not Null |
| Location | varchar(100) | Not Null |
| JobType | int | Foreign Key (JobTypes table) |
| Description | text |  |
| Requirements | text |  |

**APPLICATION TABLE**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ApplicationID | int | Primary Key, Auto Increment |
| UserID | int | Foreign Key (Users table) |
| JobID | int | Foreign Key (Jobs table) |
| ApplicationDate | datetime | Not Null |
| Status | int | Foreign Key (ApplicationStatus table) |
| Notes | text |  |
| OtherMaterials | varchar(255) |  |

**JOB TYPES TABLE**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| JobTypeID | int | Primary Key, Auto Increment |
| Type | varchar(50) | Not Null |

**APPLICATION STATUS TABLE**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| StatusID | int | Primary Key, Auto Increment |
| Status | varchar(50) | Not Null |

**SOFTWARE TESTING**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) or System Requirement Specification (SRS). System testing is an investigatory testing phase, where the focus is to have almost a destructive attitude and test not only the design, but also the behavior and even the believed expectations of the customer.

**Unit Testing** :-

Focuses on the function or a software module.

• Every module has tight security.

• You can’t enter the blank data in the database. It will throw an error.

• You can’t enter the wrong data values in the database table.

• You can’t access any report if there are no records in the table.

• You can’t enter or exit the data if you enter the wrong fields.

**Integration Testing :-**

• Now every module is connected to a one main menu directly or directly.

• The main menu can call any module. On the other hand the modules can’t call the main menu directly itself.

• Every module is easily navigated from one page to another page.

• If the page is not found or missing then the page is not found error occur.

• The links of pages are passed by the common methods which we used in several methods.

**System Testing :-**

• It checks the whole software by combining the different integrated pages.

• It also checks the hardware part for the software.

• It throws an error if the basic platform is not identified on the system to run software.

**White-box testing** :-

White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

REFERENCES

1) Software Engineering, A Practitioner’s Approach Eight Edition by Roger S. Pressman.

2) Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition.

3) [www.wikipedia.com](http://www.wikipedia.com)

4) www.geeksforgeeks.com