
SOFTWARE REQUIREMENTS SPECIFICATION

FOR

JOB SEARCH APPLICATION

Prepared By:-

JAYASHREE K

YAZHINI E

SWETHA S

SRIMATHI V

1.INTRODUCTION

1.1 Purpose:

- This document outlines the requirements for the "Naan Mudhalvan" job search using MERN.
- The Job Search Application, developed using nodejs, mongodb on the server-side React.js on the client-side, aims to simplify the job search process for job seekers and streamline the recruitment process for employers. This modern, web-based platform serves as a comprehensive, user-friendly portal connecting job seekers with employment opportunities and employers with qualified candidates.

1.2 Document Conventions:

This document serves as the introduction to the Software Requirements Specification (SRS) for the Job Search Application, which will be developed using Java on the server-side and React.js on the client-side. To ensure clarity and consistency throughout the SRS and associated project documentation, the following document conventions and notations will be adhered to

Main Titles and Subtitles:

Main Titles (Section Headers): Use clear and descriptive main titles to represent major sections of the document. These titles should be numbered for easy reference and navigation.

Subtitles (Subsections): Use subheadings to further break down topics within a section. Subtitles should be labeled with letters for hierarchy.

➤ Terminology

- ❖ **User Roles:** To refer to different user roles within the application, the following terms will be used.

- ❖ **Job Seeker:** Represents individuals seeking employment opportunities.

1.3 SCOPE FOR DEVELOPMENT PROJECT

- The scope of the "Naan Mudhalvan" Job Search Application encompasses the development of a dynamic, user-friendly web-based platform that caters to the diverse needs of job seekers and employers. This platform will facilitate job searches, applications, and employer-employee interactions while maintaining a robust and secure backend infrastructure.
- The scope of this project encompasses the development of a feature-rich job portal accessible via web browsers and compatible with various devices, including desktops, tablets, and mobile phones. The application will provide a seamless experience for both job seekers and employers, offering tools for job searching, job posting, application management, and communication.

1.4 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification, which is this document.
- UI: User Interface, referring to the graphical interface through which users interact with the application.
- API: Application Programming Interface, allowing the application to interact with external services or components.
- RDBMS: Relational Database Management System, used for storing and managing application data.
- OAuth: Open Authorization, a protocol for secure user authentication and authorization.
- SMTP: Simple Mail Transfer Protocol, for sending email notifications.
- CDN: Content Delivery Network, used to optimize content delivery.

- SQL: Structured Query Language, used for querying and managing the database.
- HTTPS: HyperText Transfer Protocol Secure, a secure version of HTTP used for encrypted data transfer over the internet.

1.5 REFERENCES

➤ BOOK

Software Engineering: A Practitioner's Approach by Roger S. Pressman: This book provides a comprehensive overview of software engineering principles, including requirements engineering and the creation of SRS documents.

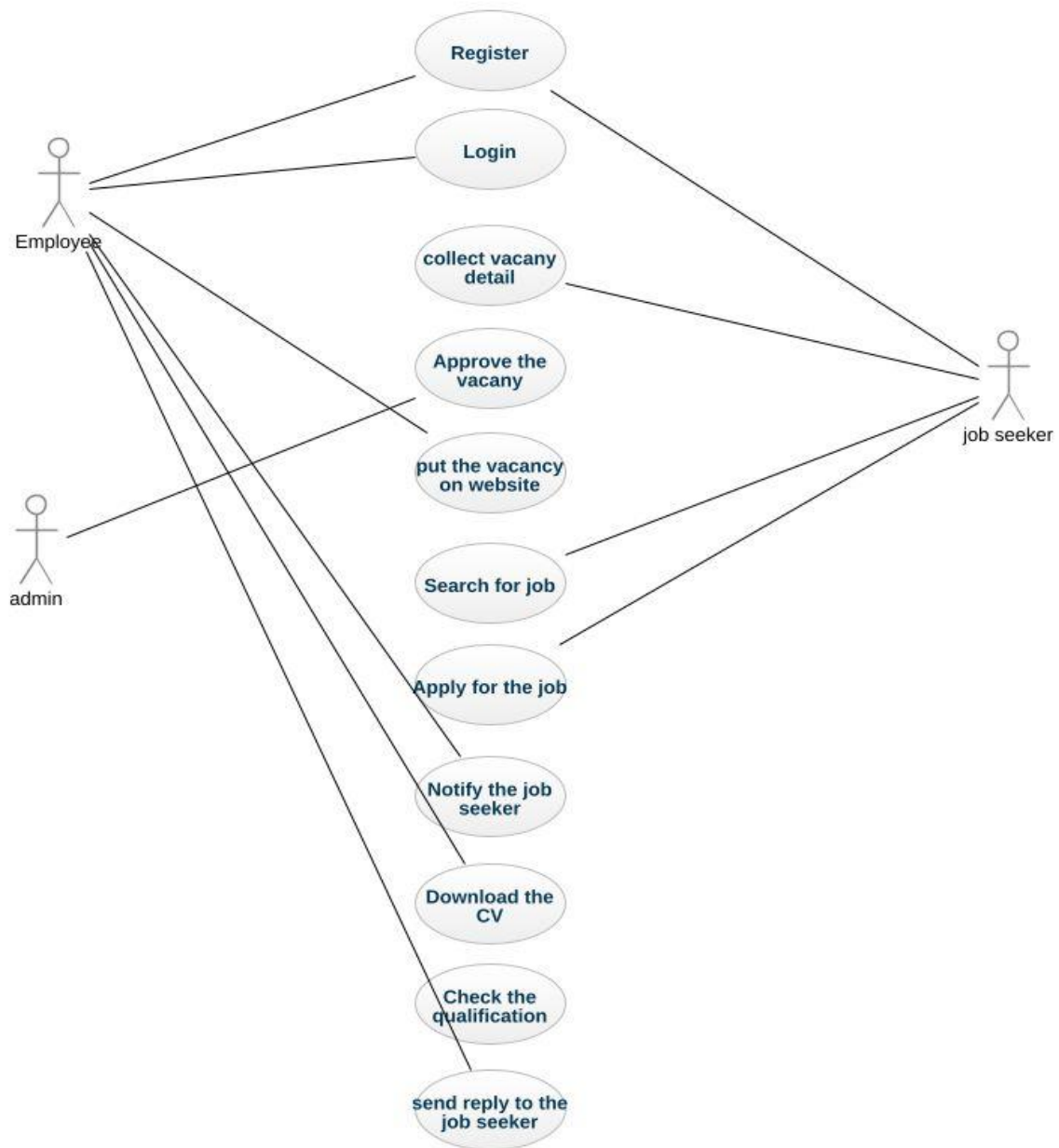
➤ WEBSITE:

https://www.mlsu.ac.in/econtents/16_EBOOK-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf

2. OVERALL DESCRIPTIONS

2.1 PRODUCT PERSPECTIVE

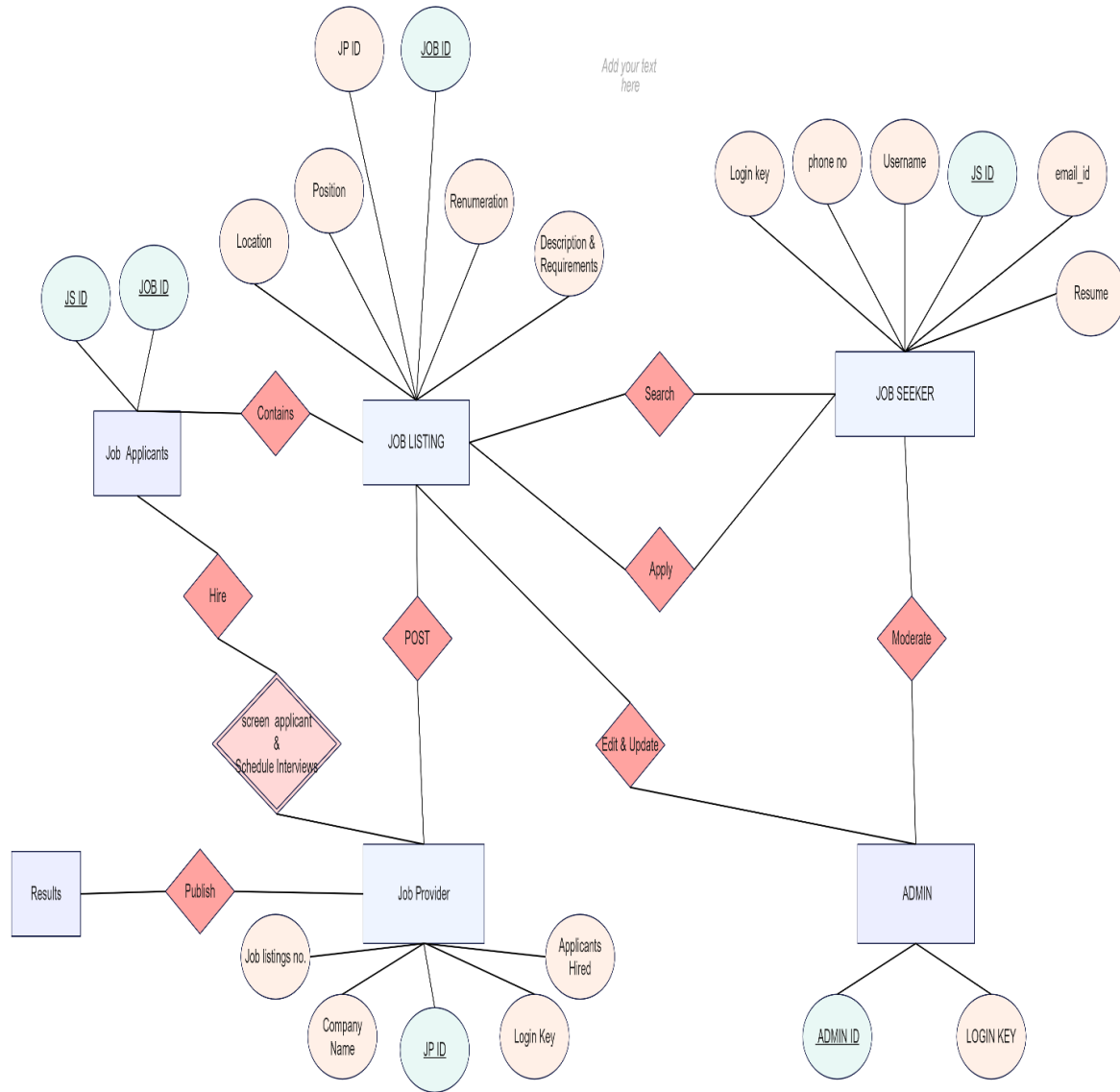
Use case diagram of Job search application



This is a broad level diagram of the project showing a basic overview. Its an platform connecting job seekers and employers, allowing job search, applications, and job posting, powered by Java backend and React frontend. The objective is Efficient job matching, seamless hiring processes, and user-friendly interface for job seekers and employers.

2.2 PRODUCT FUNCTION

Entity relationship diagram of job search application



In the ER diagram for the job search application, users can be either job seekers or employers, with various connections among them. Job postings, created by employers, are categorized by industry, location, and required skills. Job seekers submit applications for specific postings and can upload their resumes, associating them with their profiles. Companies post job vacancies and are linked to job postings, while specific skills are required for each job posting.

2.3 USER CLASSES AND CHARACTERSTICS

The job search application provides different types of services based on the type of users [Job Seeker/Employer]. Employers, acting as controllers, have administrator privileges, while job seekers can be either individuals seeking employment opportunities or professionals looking for job listings. This setup allows for distinct user roles and functionalities within the job search application, with employers having administrative authority and job seekers being the primary users.

The features that are available to the Employers are:-

- Employers can post job vacancies.
- They can categorize job listings by industry or type.
- Employers can view the list of job postings in each category.
- Employers can receive and manage job applications from job seekers.
- They have the ability to edit job posting information.
- Employers can access reports on posted jobs.
- They can access reports on job applications.
- Employers have access to profiles and accounts of job seekers.
- Employers have access to profiles and accounts of job seekers.

The Features available to Job Seekers are:-

- Job seekers can explore different job categories or industries.
- They can browse the list of job postings within each category.
- Job seekers can create and manage their profiles on the platform.
- They can view the jobs they've applied to.
- Job seekers can submit requests for specific job listings.
- They can access their job application history.

Job seekers can search for particular job listings on the platform.

2.4 OPERATING ENVIRONMENT

The Job Search Application is a web-based platform designed to operate in the Windows environment and is compatible with popular web browsers, including Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. It is optimized for compatibility with Internet Explorer 6.0 and most features function seamlessly on Mozilla Firefox and Opera 7.0 or higher versions. The only requirement for using this online product is an internet connection.

In terms of hardware configuration, the application is lightweight and can run on standard systems with a minimum of a 40 GB hard disk, a 15" color monitor, and a 122-key keyboard. The basic input devices required are a keyboard and mouse, while the primary output device is the monitor. Additional peripherals like printers are also supported for printing job-related documents and information.

2.5 ASSUMPTIONS AND DEPENDENCIES

The assumptions are:-

- The codebase is expected to be error-free and well-maintained.
- The application should prioritize user-friendliness for ease of use.
- User, job posting, and company information will be stored in a database accessible via the website.
- The system should have sufficient storage capacity and ensure fast database access.
- It should provide efficient search functionality and support rapid transactions.
- The Job Search Application operates 24/7.
- Users can access the application from any computer with internet browsing capabilities and an active internet connection.
- Users must have valid usernames and passwords for accessing their online accounts and performing actions.

The dependencies are:-

- The product relies on specific hardware and software configurations for its functionality.
- The project's development and operation are based on defined listing requirements and specifications.
- The end users, including administrators, should have a proper understanding of the application's functionality.
- The system should maintain a comprehensive set of general reports.
- User information is stored in a database accessible by the Job Search Application.
- Any updates related to job listings and applications are recorded in the database, and the entered data must be accurate.

2.6 REQUIREMENTS

Software Configuration:-

- Front-end development is primarily done using React, a JavaScript library.
- The application is designed to be compatible with various modern web browsers, such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
- nodejs and mongodb may be utilized for certain server-side components.
- Integrated development environments (IDEs) like Visual Studio Code or IntelliJ IDEA can be used for front-end and Java development.
- The application's back end is powered by nodejs, and database mongodb is used for data storage.

Hardware Configuration:-

- Processor: A modern multi-core processor, such as an Intel Core i5 or equivalent.
- Hard Disk: Adequate storage capacity, typically 128GB or more.

- RAM: A minimum of 4GB RAM is recommended for smooth application performance. The choice of the operating system depends on the development and deployment environment, but it should support React and Java development.

2.7 DATA REQUIREMENT

- Inputs include user interactions such as job searches, job applications, and account management actions.
- Outputs consist of search results, application statuses, and user account details, including job application history.
- User inputs can include actions like creating a job seeker account, browsing job listings, and applying for jobs.
- Output includes information like the timestamp, date, and a list of current job applications in the user's account when requested from the server.

3. EXTERNAL INTERFACE REQUIREMENTS:

3.1 User Interfaces:

GUI (Graphical User Interface):

- The application should provide an intuitive and visually appealing graphical interface for both users and administrators.
- It should enable users to perform tasks like job search, application submission, and profile management.
- Quick reports on job applications and activity within a specific time frame should be accessible.
- A search feature should allow users to filter job listings based on various criteria.
- All modules within the application should seamlessly integrate into the graphical user interface while maintaining design simplicity and consistency.

Login Interface:

Users can log in using their registered credentials. If incorrect information is provided, an error message should be displayed.

Job Search:

- New users have the option to register by providing their details.
- Users can search for jobs by specifying criteria such as job type, location, and keywords

Categories View

A view that displays job categories should be available, allowing administrators to manage and customize categories as needed.

User Dashboard:

The user dashboard provides essential functionalities for both job seekers and employers, including managing user profiles, job listings, and applications.

Admin Control Panel:

- An admin control panel should allow administrators to manage user accounts, job listings, and application options.

Standardized Design:

- The application's design should follow a simple and consistent template across different

User Management Integration:

- The user interface should seamlessly integrate with the user management module, with a dedicated section for user login and logout.
- Please note that this is a simplified adaptation, and additional details and features would be required for a complete job search application in React.js.

4. SYSTEM FEATURES

4.1 SYSTEM FEATURE 1

4.1.1 DESCRIPTION AND PRIORITY

It offers a user-friendly interface for job posting, application submission, and communication between employers and candidates ,and manage profiles, search for jobs using various filters, receive real-time notifications, and access support resources. The priority of this job search application is high, as it addresses a fundamental need in the job market, helping job seekers find opportunities and employers connect with potential hires.

4.1.2 STIMULUS/RESPONSE SEQUENCES

A user opens the application and clicks on the "Login" button. The application presents a login form where the user can enter their credentials or choose to sign up for a new account. Upon successful login, the user is directed to their dashboard.

4.1.3 FUNCTIONAL REQUIREMENTS

Typically, software engineers create and apply functional requirements to software during the development stages of a project to ensure their software is easy to use and operational. Functional requirements can vary in behaviors, features and protocols, depending on the user's industry. For example, a game designer, may use different functional requirements in software that focus on game design, while a teacher may use functional requirements that focus on student usability.

4.2 SYSTEM FEATURE 2

4.2.1 USER REGISTRATION AND PROFILES:

The application should allow users to easily create accounts with essential information and build comprehensive profiles, including details about their work history, qualifications, and skills. Users should have the flexibility to edit and update their profiles as needed. User Registration and Profiles.

4.2.2 JOB LISTINGS AND SEARCH:

A central database of job listings and a robust search functionality are crucial. Users should be able to efficiently search for job opportunities using filters based on location, industry, salary range, and other relevant criteria. The application should provide a user-friendly interface for exploring job listings.

5. OTHER NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENT

The job portal application, built using React for the frontend and Java for the backend, must meet stringent performance requirements to ensure a seamless user experience.

- It should maintain a rapid response time, with pages loading in under three seconds, even during peak usage, supporting a high concurrency of users.
- The system must scale horizontally to accommodate increasing traffic, optimize database queries for swift data retrieval, and implement efficient caching mechanisms.
- Security measures should not compromise speed, and resource utilization should be efficient.
- Regular load testing, monitoring, and code optimization are vital, as is the use of Content Delivery Networks for static assets.

5.2 SAFETY REQUIREMENT

Safety requirements for a job portal application developed using React and Java are crucial to safeguard user data and ensure secure interactions. The application must adhere to stringent data protection standards, including encryption of sensitive information like user credentials and personal data.

- It must comply with relevant data privacy regulations, such as GDPR or HIPAA, depending on the scope of data handling.

- To enhance safety, user-generated content should be monitored for inappropriate or harmful content, and reporting mechanisms should be available to users.
- Finally, the application must have a robust disaster recovery and backup strategy to prevent data loss and ensure business continuity in case of unforeseen incidents.

5.3 SECURITY REQUIREMENT

The application must employ strong encryption protocols to secure sensitive user information, such as login credentials and personal data.

- Robust authentication and authorization mechanisms should be implemented to ensure that users can only access the data and features appropriate to their roles.
- Input validation must be stringent to guard against common web vulnerabilities like SQL injection and cross-site scripting (XSS).
- Additionally, the application should have mechanisms to detect and respond to security incidents, such as intrusion detection systems and log monitoring.
- Compliance with data protection regulations (**e.g., GDPR**) and secure coding practices should be enforced throughout development.

5.4 REQUIREMENT ATTRIBUTES

- **Functional Requirements:** Clearly define the features and capabilities the application must possess, such as user registration, job posting, search filters, application submission, and user profiles.
- **Performance Requirements:** Specify criteria for response times, page load times, and database query speeds to ensure the application operates efficiently, even during peak usage.

- **Security Requirements:** Outline measures to protect user data, including encryption, authentication, authorization, and vulnerability assessments to safeguard against cyber threats.
- **Documentation and Training Requirements:** Detail the need for user manuals, developer documentation, and training materials to assist both end-users and development teams.
- **Cost and Resource Requirements:** Identify budget constraints, hardware and software resources, and personnel needed to develop, deploy, and maintain the application.

5.5 BUSINESS RULES

- Users must register with valid information.
- Employers can post jobs meeting certain criteria.
- Users can search for jobs and apply filters.
- Define pricing models and payment processes.
- Users receive notifications and can communicate.
- Users can seek help and access customer support.

5.6 USER REQUIREMENT

- Allow users to create accounts easily.
- Enable users to search for jobs based on criteria like location, industry, and salary.
- Allow users to apply for jobs with minimal effort.
- Provide timely updates on job applications and relevant job listings.
- Offer a secure messaging system for communication with employers.
- Allow users to save and organize favorite job listings.
- Let users control the visibility of their profiles and personal information.
- Allow users to rate and review employers.
- Provide easy access to customer support and assistance.

- Ensure the app is accessible and works well on mobile devices.

6. OTHER REQUIREMENTS

6.1 DATA AND CATEGORY REQUIREMENT

Each job listing should display relevant information, including job titles , company name, location, job description, required qualifications, and application details .Job seekers can filter job listings by category (IT, Healthcare, Finance, etc.) to find relevant positions .Employers can enter and update job details in a standardized format when posting listings .Admins can access and manage all job listings and user accounts.

6.2 APPENDIX

A: Administrator Roles, Abbreviations, Acronyms, Assumptions **B:** Job Listings, Business Rules **C:** Classes, Clients, Conventions **D:** Data Requirements, Dependencies **G:** Graphical User Interface (GUI) **K:** Keys **L:** Library, Librarian **M:** Members **N:** Non-Functional Requirements **O:** Operating Environment **P:** Performance, Perspective, Purpose **R:** Requirements, Requirement Attributes **S:** Safety, Scope, Security, System Features **U:** Users, User Classes and Characteristics, User Requirements.

6.3 GLOSSARY

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A user with administrative privileges for the application.
- Client: Refers to the intended users of the job search application.
- SQL: Abbreviation for Structured Query Language, used to retrieve and manipulate data from a database.
- SQL Server: A database management system used to store and manage data efficiently.
- Layer: Represents a distinct section or component within the application.

- User Interface Layer: The part of the application that users directly interact with.
- Application Logic Layer: The component responsible for processing and computations on the web server.
- Data Storage Layer: The segment of the application responsible for storing and managing data.
- Use Case: A high-level diagram illustrating the fundamental functionalities and interactions within the job search application.
- Class Diagram: A static structure diagram that depicts the relationships, attributes, and structure of various components (e.g., classes) within the system.
- Interface: A means of communication and interaction between different parts of the application.
- Unique Key: A field or attribute used to uniquely identify and differentiate entries within a database.

6.3 CLASS DIAGRAM

- A class in the context of the Job Search Application is an abstract, user-defined representation of a data type. It defines the attributes of the data and specifies the operations that can be performed on objects of that data type. Each class has a name, a set of attributes describing its properties, and a set of methods representing its functionalities. The static model of the application consists of these classes and their relationships.
- Within the application, several main classes are central to its functionality, and they are related to other classes necessary for their operation. Various types of relationships exist between these classes, including normal associations, aggregations, and generalizations.

