



SOFTWARE DESIGN SPECIFICATION

ReactJs  
  
Job Portal

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Rushikesh Mandhare | **Approved By:** | Harshada Topale |
| **Created On:** | 01-07-2024 | **Approved On:** | 01-07-2024 |

Page left blank intentionally

**INDEX**

[**1** **PURPOSE** 2](#_Toc142418236)

[**2** **PROJECT SCOPE** 2](#_Toc142418237)

[**3** **SYSTEM OVERVIEW** 2](#_Toc142418238)

[**4** **DESIGN CONSIDERATIONS** 2](#_Toc142418239)

[4.1 Requirements 3](#_Toc142418240)

[4.2 Assumptions 3](#_Toc142418241)

[4.3 Dependencies 3](#_Toc142418242)

[**5** **SYSTEM ARCHITECTURE** 3](#_Toc142418243)

[5.1 Architectural Strategies 4](#_Toc142418244)

[5.2 Structure & Relationships 4](#_Toc142418245)

[**6** **DETAILED DESCRIPTION OF COMPONENTS** 4](#_Toc142418246)

[**7** **INTEGRATION** 5](#_Toc142418247)

[**8** **APPENDICES** 1](#_Toc142418248)

[8.1 Appendix A – Detailed Description of Components 1](#_Toc142418249)

**General Instructions for using the Live Project POC Document**

* This template and the subsequent document created using this template is a confidential document and is the intellectual property of Cloud Counselage Pvt. Ltd. Circulating it outside of the organisation without the consent of Cloud Counselage Pvt. Ltd. is the breach of company policies and will lead to legal actions
* The Design Specification of a software forms the basis of development of software
* The **text between inequality (< >) is to be replaced** by relevant text
* Please **remove the yellow highlight on the Text** between the inequality (< >). This is done to help you notice the text to be changed/replaced
* The text in *italics* highlighted in grey is just for reference and should be removed after adding the relevant text

# **PURPOSE**

The purpose of this Software Design Specification (SDS) document is to provide a comprehensive blueprint for the development of a professional networking platform. This document will outline the project's scope, system overview, design considerations, system architecture, detailed description of components, integration strategies, and appendices, ensuring that all stakeholders have a clear understanding of the project.

# **PROJECT SCOPE**

The project aims to develop a professional networking platform similar to LinkedIn, designed to help users find jobs or internships, connect with professionals, and learn new skills. The platform will include features such as user registration, profile management, job postings, connections, messaging, notifications, and a news feed. The application will be built using modern web technologies and deployed on a scalable infrastructure.

# **SYSTEM OVERVIEW**

# The system will consist of a web-based application accessible via major web browsers. Users will be able to create and manage profiles, search for jobs and connections, send and receive messages, post and view job listings, and receive notifications. The backend will handle data storage, processing, and API integrations, while the frontend will provide an intuitive user interface.

# **DESIGN CONSIDERATIONS**

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

## Requirements

* **Functional Requirements:**

1. User registration and authentication
2. Profile creation and management
3. Job postings and search functionality
4. Connection requests and management
5. Messaging system
6. Notifications for user activities
7. News feed displaying user activities and updates

* **Non-functional Requirements:**

1. Performance: The system should handle at least 1000 concurrent users.
2. Security: **Data must be securely stored and transmitted.**
3. **Usability: The platform should be user**-friendly and intuitive.
4. Scalability: The system should be able to scale with increasing user base.

## Assumptions

* Users will have internet access and basic familiarity with social networking platforms.
* The platform will handle moderate traffic initially, with the potential to scale.
* Users will provide accurate and truthful information in their profiles.

## Dependencies

* Integration with external job listing APIs.
* Email services for notifications and verifications.
* OAuth providers for user authentication.

# **SYSTEM ARCHITECTURE**

## Architectural Strategies

* Frontend: Built with React.js to ensure a responsive and dynamic user interface.
* Backend: Implemented using Node.js with Express to handle server-side logic and APIs.
* Database: MongoDB for scalable and flexible data storage.
* Hosting: Deployed on Heroku/GitHub Pages for accessibility and scalability.

## Structure & Relationships

* Frontend and Backend Communication: Use RESTful APIs to communicate between the frontend and backend.
* Database Management: MongoDB will store user profiles, job postings, messages, and other data.
* External Services: Integrate with email services for notifications and OAuth providers for authentication.

# **DETAILED DESCRIPTION OF COMPONENTS**

* User Authentication: Handles user registration, login, and secure access using JWT.
* Profile Management: Allows users to create and update their profiles, including professional information, skills, and interests.
* Job Postings: Enables users and businesses to post job openings and allows users to search and apply for jobs.
* Connections: Manages user connections, including sending and accepting connection requests.
* Messaging: Provides a real-time messaging system for user communication.
* Notifications: Sends notifications for various user activities, such as connection requests and job applications.
* News Feed: Displays updates and posts from connections, job recommendations, and other relevant information.

# **INTEGRATIONS**

* API Integration: Regularly test and maintain integrations with external job listing APIs, email services, and OAuth providers.
* Continuous Integration and Deployment: Use CI/CD pipelines to automate testing and deployment processes.
* Monitoring and Logging: Implement monitoring tools to track system performance and log errors for debugging.

# **APPENDICES**

## Appendix A – Detailed Description of Components

1.

|  |  |
| --- | --- |
| **Identification** | **UserAuthenticationModule** |
| **Type** | Module |
| **Purpose** | Handles user registration, login, and JWT-based authentication. Ensures secure handling of user credentials and session management. |
| **Subordinates** | RegistrationHandler<br>- LoginHandler<br>- JWTGenerator |
| **Dependencies** | |  | | --- | |  |  |  | | --- | | Relies on the DatabaseModule for storing user credentials.<br>- Uses the EmailServiceModule for sending verification emails.<br>- Interacts with the UserInterfaceModule for user actions. | |
| **Interfaces** | API endpoints for registration, login, and email verification. |
| **Resources** | Requires a MongoDB database instance.<br>- Needs an SMTP server for sending email notifications. |
| **Processing** | Manages the registration and login processes, validates user credentials, and generates JWT for authenticated sessions. |
| **Data** | User credentials such as username, password, and email are stored and validated against the database. |

2.

|  |  |
| --- | --- |
| **Identification** | **ProfileManagementModule** |
| **Type** | |  | | --- | | Module |  |  | | --- | |  | |
| **Purpose** | Manages user profiles, allowing users to create and update their professional information, skills, and interests |
| **Subordinates** | ProfileCreationHandler<br>- ProfileUpdateHandler<br>- ProfileViewHandler |
| **Dependencies** | Depends on the DatabaseModule for storing and retrieving profile data.<br>- Interacts with the UserInterfaceModule for user actions. |
| **Interfaces** | API endpoints for creating, updating, and retrieving profiles. |
| **Resources** | Requires a MongoDB database instance. |
| **Processing** | Handles the creation, update, and retrieval of user profiles. |
| **Data** | User profile data including professional information, skills, and interests. |

3.

|  |  |
| --- | --- |
| **Identification** | **JobPostingsModule** |
| **Type** | |  | | --- | | Module |  |  | | --- | |  | |
| **Purpose** | Allows businesses to post job openings and users to search and apply for jobs. |
| **Subordinates** | JobPostingHandler<br>- JobSearchHandler<br>- JobApplicationHandler |
| **Dependencies** | Depends on the DatabaseModule for storing and retrieving job postings.<br>- Interacts with the UserInterfaceModule for user actions. |
| **Interfaces** | API endpoints for posting jobs, searching jobs, and applying for jobs. |
| **Resources** | Requires a MongoDB database instance. |
| **Processing** | Manages job postings from businesses, enables job search with filters, and processes job applications. |
| **Data** | Job postings data including job title, description, requirements, and application details. |