

**Software Requirements Specification (SRS)**

Project: RECRUITMENT SOLUTION AGENCY (RSA)

Yazan Mohammed Amin

|  |  |  |
| --- | --- | --- |
| **NAME** | **CONTACTNO** | **EMAIL** |
| Yazan Mohammad Amin | 0790939430 | [Yznbrhm77@gmail.com](mailto:Yznbrhm77@gmail.com) |

**Cover Page**

**Revisions**

**Table of Contents**

**Table of contents**

**1 INTRODUCTION…………………………………………………...…3**

1.1 Product Overview…………………………………………………….…………...3

1.2 Purpose……………………………………………………………….…………...3

1.3 Scope …………………………………………………………………………3

**2 Specific Requirements……………………………………………...…...4**

2.1 External interface requirements…………………………………………...……5

2.1.1 User interfaces……………………………………………………………………………6

2.1.2 Hardware Interfaces………………………………………………………………6

2.1.3 Software Interfaces………………………………………………………………7

2.3.4 Communications Interfaces………………………………………………………………7

2.2 Software Product Features……………………………………………………………8

2.3 Software System Attributes……………………………………………………………8

2.3.1 Reliability……………………………………………………………9

2.3.2 Availability……………………………………………………………9

2.3.3 Security……………………………………………………………9

2.3.4 Maintainability……………………………………………………………10

2.3.5 Portability…………………………………………………………10

2.3.6 Performance……………………………………………………………10

2.4 Database Requirements ……………………………………………………………11

**3. Use case diagrams and specifications……………………….12**

3.1 Use Case Name…………………………………………………………13

3.2 Identifier………………………………………………………………13

3.3 Description………………………………………………………………14

3.4 Preconditions………………………………………………………………15

3.5 Actors……………………………………………………...16

3.6 Main flow………………………………………………….…….16

3.7 Alternative flow……………………………………………………...16

**4 ADDITIONAL MATERLAL……………………………………17**

**1 INTRODUCTION**

SRS stands for Software Requirement Specifications. In this document we will describe about the requirements of the whole system which we are going to design for our client. This will include information about scopes, constrains, limitations, functional and non-functional requirements, apart from that this document will focus on the system that how it will interact with other requirements. This will help us to clear all the issues between client and development. As this document is completely based on the requirements, this is really important that our designers and client should be very clear about the final product, any misunderstandings or issues can be easily resolved at this stage with the help of SRS.

**1.1 Product Overview**

Steps to summarize the goals of the project and the programs that will be delivered User Interfaces and Tool Interfaces and Software Interfaces and Security and Additional Items.

**1.2 Purpose**

**The main purpose of this document is:**

1. Describe various components of the system.
2. Information about what system is supposed to achieve and how.
3. Details of the functional and non-functional requirements.
4. Quick view of entities and their relationships.
5. Information about different level of DFD's.
6. How system interact with user and within entities.
7. Outline the modules and parameters in structural decomposition diagrams.

**1.3 Scope**

SRS is the document which we make requirements more clear and defined. SRS will give exact information to the designers and developers about the system required by the client. It will bemore technical rather than verbal information. Scope: SRS plays very important role in the development of our system. It defines all the requirements of all the three different levels of our system – Admin, Employer- job Secker.

1. **Specific Requirements**

**1. User Registration and Login:**

* The system should allow users to register with their email and create a password.
* Registered users should be able to log in securely with their credentials.

**2. Candidate Management:**

* The system should provide functionalities to add, update, and delete candidate profiles.
* Candidate profiles should include information like name, contact details, work experience, skills, etc.

**3. Job Posting and Management:**

* Recruiters should be able to post new job openings with details such as job title, description, required skills, and location.
* Recruiters should be able to update or close existing job postings.

**4. User Interface:**

* The platform should have an intuitive and user-friendly interface for easy navigation and usage.

**5. Performance and Scalability:**

* The system should be designed to handle a large number of candidates, job postings, and users efficiently.

**2.1 External interface requirements**

**1. User Interface (UI):**

* The system should have a user-friendly web-based interface accessible through popular web browsers.
* The UI should be responsive and adaptable to different screen sizes (desktop, tablet, mobile).

**2. Registration and Login:**

* Users should be able to register and log in using a standard web form.
* The registration form should capture necessary user details like name, email, and password.

**3. Job Posting and Application:**

* Recruiters should have an interface to post new job openings with fields for job title, description, skills required, location, etc.
* Candidates should be able to view job postings and apply through the platform.

**4. Database Interaction:**

* The system should interact with a backend database to store and retrieve user data, job postings, and application records.
* Security Interfaces:
* The system should employ secure communication protocols (e.g., HTTPS) to protect user data during transmission.
* APIs should be secured with authentication and authorization mechanisms to prevent unauthorized access.

**5. Backup and Restore:**

* The system should have mechanisms to perform regular data backups and support data restoration in case of failures.

**2.1.1 User interfaces**

**1. Job Posting Interface:**

* Recruiters have an interface to create and manage job postings.
* They can input job details, such as job title, description, required skills, location, and application deadline.

**2. Candidate Profile Interface:**

* Candidates have an interface to create and manage their profiles.
* They can update personal information, work experience, education, skills, and upload their resume.

**3. Job Search Interface:**

* Candidates can search for job openings based on various criteria like job title, skills, location, etc.
* The interface displays search results and provides the option to view job details and apply.

**2.1.2 Hardware Interfaces**

**1. Printers and Scanners (Optional):**

* The system might need to interface with printers and scanners if there are requirements for generating physical documents (e.g., contracts, offer letters).

**2. Mobile Devices:**

* If the system has a mobile app, it needs to interact with mobile devices' hardware components such as touchscreens, cameras, and GPS.

**3. Email Servers:**

* The software system will need to interact with email servers to send email notifications to users.

**2.1.3 Software Interfaces**

**1. Database Interface:**

* The system will interact with a database management system to store and retrieve data such as user profiles, job postings, candidate applications, etc.
* The database interface allows the application to perform CRUD operations (Create, Read, Update, Delete) on the database.5.

**2. Authentication and Authorization Interface:**

* The system will have an interface for user authentication and authorization.
* It verifies user credentials during login and determines the user's access rights and permissions.

**3. Application Tracking Interface:**

* This interface handles the tracking of candidate applications for job postings.
* It allows recruiters to review and manage candidate applications for specific job positions.

**2.3.4 Communications Interfaces**

**1. Protocol HTTP / HTTPS:**

* The system uses the Hypertext Transfer Protocol (HTTP) protocol or its version for secure HTTPS communication between the client (web browser) and the server.
* HTTP / HTTPS is used for data transfer for user registration and login registration and job announcements, candidate definition files and interactions.

**2. Protocol MTP:**

* The system uses the Simple Mail Transfer Protocol (SMTP) to send email messages to users.
* SMTP is used to activate the account and e-mail, to reset the password, and to update the application status and other notifications.

**3. Push Notifications:**

* If the system contains a mobile application, it may perform payment notification services to send real-time notifications to mobile devices.
* Push notifications Allow users to receive updates even when the application is not actively in use.

**2.2 Software Product Features**

**1. Job Posting and Management:**

* Allow recruiters to post new job openings with job title, description, skills required, location, and application deadline.
* Enable recruiters to update or close existing job postings.

**2. Job Search and Filtering:**

* Allow candidates to search and filter job openings based on criteria like job title, skills, location, etc.
* Provide advanced search options for precise job matching.

**3. Application Tracking:**

* Allow candidates to apply for job positions through the platform.
* Enable recruiters to track and manage candidate applications for their job postings.
* Send automated application status updates to candidates.

**2.3 Software System Attributes**

**1. Security:**

* Implement robust authentication and encryption mechanisms to protect user data and prevent unauthorized access.
* Conduct regular security audits and vulnerability assessments.

**2. Scalability:**

* Design the system to handle a large number of users, job postings, and applications efficiently.
* Use scalable infrastructure and technologies to support future growth.

**3. Reliability:**

* Ensure high availability and minimal downtime through redundant server configurations and fault-tolerant architecture.
* Implement automated backup and recovery procedures for data protection.

**2.3.1 Reliability**

**1. Data Integrity:**

* Employ data validation and verification techniques to maintain data integrity and prevent data corruption.

**2. Monitoring and Logging:**

* Implement real-time monitoring and logging to detect issues promptly, track system performance, and gather useful insights for improvements.

**3. Backup and Recovery:**

* Establish regular data backups to safeguard against data loss, and have a well-defined data recovery plan in case of system failures.

**2.3.2 Availability**

**1. Failover Mechanisms:**

Implement failover mechanisms to automatically switch to backup systems in case of server failures or hardware issues.

**2. Server Uptime and Maintenance Planning:**

Schedule routine maintenance during periods of low user activity to minimize service interruptions.

**3. Distributed Data Centers:**

Consider using geographically distributed data centers to improve availability and reduce the impact of regional outages.

**2.3.3 Security**

**1. Authentication and Authorization:**

* Implement secure authentication methods (e.g., multi-factor authentication) to verify user identities, and use role-based access control to define user permissions.

**2. Secure Communication:**

* Use SSL/TLS encryption to secure data transmission between the server and clients, preventing eavesdropping and data interception.

**3. Security Updates and Patch Management:**

* Keep software components, frameworks, and libraries up to date with the latest security patches to mitigate known vulnerabilities.

**2.3.4 Maintainability**

**1. Modularity and Code Organization:**

* Design the system with a modular architecture and well-organized codebase to promote code reusability and ease of maintenance.

**2. Documentation:**

* Provide comprehensive documentation for the code, APIs, and system architecture to aid developers in understanding the system's functionality and internal workings.

**3. Version Control:**

* Utilize version control systems (e.g., Git) to track changes and facilitate collaboration among developers, enabling easy rollback and identification of code modifications.

**2.3.5 Portability**

**1. Cross-Platform Compatibility:**

* Develop the application using cross-platform technologies and frameworks to ensure compatibility with various operating systems (Windows, mac OS, Linux).

**2. Responsive Design:**

* Implement responsive design principles to ensure the user interface adapts to different screen sizes and resolutions, including mobile devices.

**3. Browser Compatibility:**

* Ensure the application is compatible with multiple web browsers, such as Chrome, Firefox, Safari, and Edge, to accommodate users' preferences.

**2.3.6 Performance**

**1. Database Optimization:**

* Optimize database queries, use indexes, and minimize redundant data to improve database performance.

**2. Load Testing and Performance Monitoring:**

* Conduct load testing to assess system behavior under high user traffic and monitor system performance in real-time.

**3. Image Compression:**

* Compress and optimize images to reduce page load times and improve overall application performance.

2.4 Database Requirements

**1. Relational Database Management System (RDBMS):**

* Select a reliable and scalable RDBMS, such as MySQL, PostgreSQL, or Microsoft SQL Server, to handle structured data efficiently.

**2. Database Schema Design:**

* Design a well-structured and normalized database schema to ensure data integrity and minimize redundancy.
* Define appropriate tables to store candidate information, job postings, user credentials, application records, etc.

**3. Candidate and Job Postings Data:**

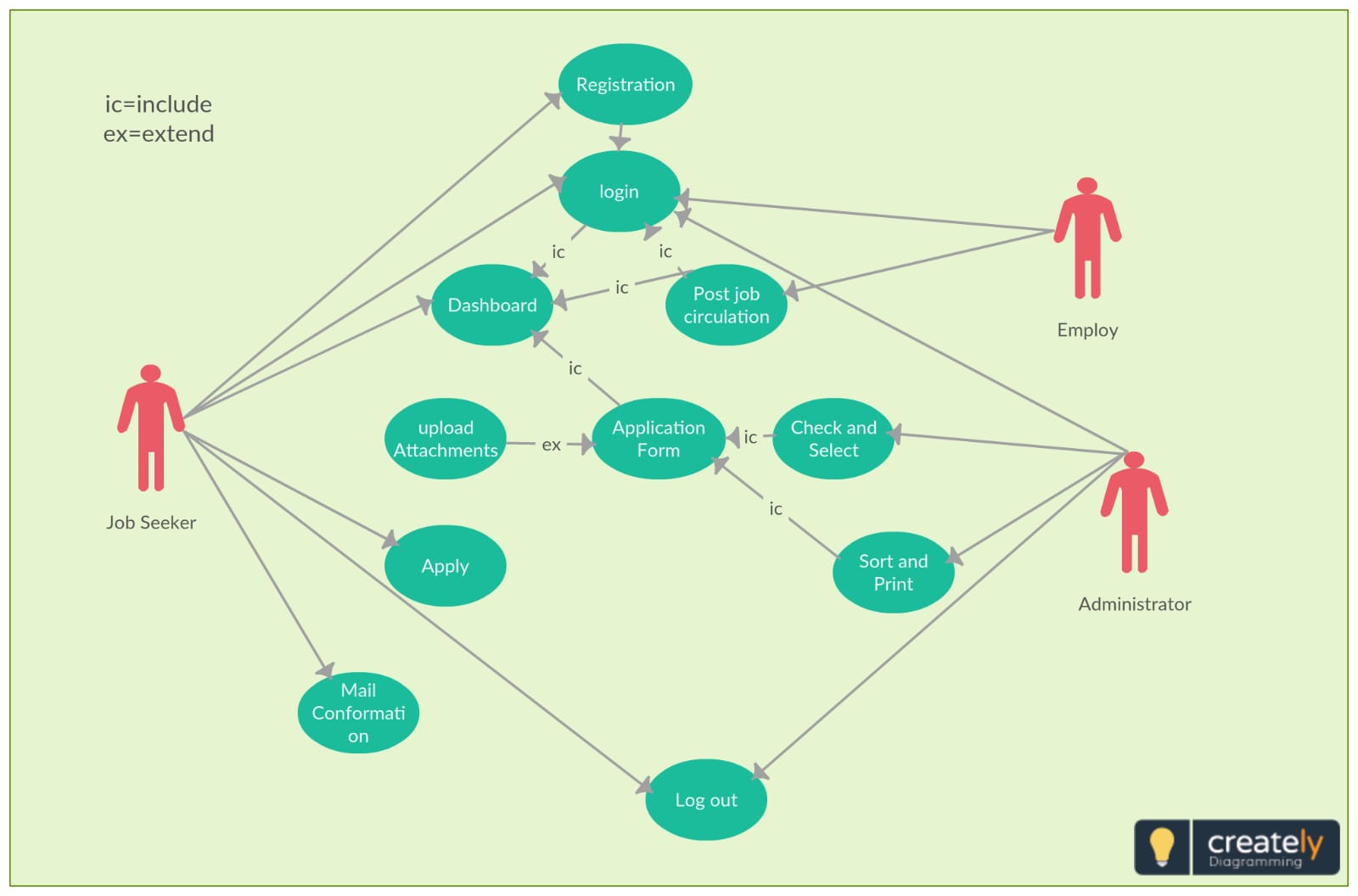
* Store candidate details, including name, contact information, work experience, education, and skills.
* Save job posting information like job title, description, required skills, location, and application deadline.

**4. User Authentication and Authorization:**

* Store user credentials securely, utilizing techniques like password hashing and salting.
* Implement roles and permissions to manage user access and ensure data security.

**5. Application Tracking and Status:**

* Design tables to track candidate applications, their associated job postings, and application status (e.g., submitted, under review, accepted, rejected).

**3. Use case diagrams and specifications**

**3.1 Use Case Name**

**1. Register Account:** RegAcc

**2. Login:** LogIn

**3. Post Job Opening:** PostJob

**4. Apply for Job:** ApplyJob

**5. View Applications:** ViewApps

**6. Update Candidate Profile:** UpdateProf

**7. View Analytics:** ViewAnalytics

**8. Reset Password:** ResetPass

**3.2 Identifier**

**1. Register Account:** UC01

**2. Login:** UC02

**3. Post Job Opening:** UC03

**4. Apply for Job:** UC04

**5. View Applications:** UC05

**6. Update Candidate Profile:** UC06

**7. View Analytics:** UC07

**8. Reset Password:** UC08

**3.3 Description**

1. **Register Account:**

* This use case enables a candidate to create a new account on the "Recruitment Solution Agency" platform. The candidate provides essential details such as their name, email address, and password to register. Once the registration is complete, an activation email is sent to the provided email address for account verification.

**2. Login:**

* This use case allows registered users (candidates, recruiters, and administrators) to log in to their respective accounts. Users must provide their email and password to access their personalized dashboards, where they can perform various actions based on their roles.

**3. Post Job Opening:**

* This use case enables a recruiter to create and post a new job opening on the platform. The recruiter provides job details such as job title, description, required skills, location, and application deadline. The job posting becomes visible to candidates once published.

**4. Apply for Job:**

* This use case allows candidates to apply for a specific job opening listed on the platform. Candidates must be logged in and meet the job's requirements. They can submit their application, including their resume and any additional information requested by the recruiter.

**5. View Applications:**

* This use case enables a recruiter to view and manage candidate applications for a specific job opening. The recruiter can review candidate profiles, resumes, and other application materials. They can then accept or reject applications based on the job's criteria.

**6. Update Candidate Profile:**

* This use case allows candidates to update their profile information on the platform. Candidates can modify details such as contact information, work experience, educational background, and skills to ensure their profiles are up to date.

**7. View Analytics:**

* This use case enables recruiters and administrators to view analytics and reports related to the platform's activities. Analytics may include data on the number of job postings, candidate applications, application status, and other relevant metrics.

**8. Reset Password:**

* This use case allows users to request a password reset in case they forget their password. Users provide their registered email address, and a password reset link is sent to that email address to initiate the password recovery process.

**3.4 Preconditions**

**1. Register Account:**

* The candidate's email address is not already registered in the system.
* The candidate provides all the required information (name, email, password) in the registration form.

**2. Login:**

* The user has a registered account in the system.
* The provided login credentials (email and password) are correct and match the account.

**3. Post Job Opening:**

* The recruiter is logged in and has a registered account in the system.
* The required information for the job posting (e.g., job title, description, location) is provided.

**4. Apply for Job:**

* The candidate is logged in and has a registered account in the system.
* The job posting is active and within the application deadline.

**5. View Applications:**

* The recruiter is logged in and has a registered account in the system.
* The job posting for which applications are to be viewed exists and is created by the same recruiter.

**6. Update Candidate Profile:**

* The candidate is logged in and has a registered account in the system.
* The candidate's profile exists in the system.

**7. Update Job Posting:**

* The recruiter is logged in and has a registered account in the system.
* The job posting to be updated exists and is created by the same recruiter.

**8. Close Job Posting:**

* The recruiter is logged in and has a registered account in the system.
* The job posting to be closed exists and is created by the same recruiter.
* The application deadline for the job posting has not passed.

**9. Delete Candidate Profile:**

* The candidate is logged in and has a registered account in the system.
* The candidate's profile exists in the system.

**10. Delete Job Posting:**

* The recruiter is logged in and has a registered account in the system.
* The job posting to be deleted exists and is created by the same recruiter.
* The job posting is not already closed.

**3.4 Actors**

a. Jab Seeker.

b. Employ.

c. Administrator.

**3.6 Main flow**

1. The candidate accesses the registration page.

2. The system presents the candidate with a registration form.

3. The candidate fills in the required details (name, email, password).

4. The candidate submits the registration form.

5. The system validates the input data and checks for existing accounts with the same email.

6. If the email is not already registered, the system creates a new candidate account and sends an activation email to the provided email address.

7. The candidate activates the account by clicking the activation link in the email.

**3.7Alternative flow**

If the candidate enters an email address that is already registered, the system displays an error message and prompts the candidate to provide a different email address.

**4 ADDITIONAL MATERLAL**

**1. System Architecture Diagram:**

* An architectural overview of the software system, illustrating the components and their interactions.

**2. Performance Testing Reports:**

* Reports detailing the results of performance testing, including response times, throughput, and resource usage.

**3. Security Plan and Audit Reports:**

* Documentation outlining the security measures implemented in the system and the results of security audits.

**4. Deployment Plan:**

* Guidelines and instructions for deploying the software system on different environments (development, staging, production).

**5. Project Timeline and Milestones:**

* A schedule outlining the project's phases, milestones, and delivery dates.