Software Requirements Specification

ResumeRover: Intelligent Resume Screening System

Version 1.0 - Prepared by: Ashidu Dissanayake

Version 1.1 - Sehan Dissanayake Dineth Randula Shanil Praveen Yugan Ekanayaka- Refined the functional and nonfunctional requirements relevant to the AI Candidate Ranking

Version 1.2 Dineth RandulaRefined accuracy

Date: April 10, 2025

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of ResumeRover, an Intelligent Resume Screening System. It outlines the functional and nonfunctional requirements, system interfaces, and constraints of the software system. This document serves as a reference for the development team and as a contract between stakeholders and developers.

1.2 Document Conventions

This document follows IEEE Standard 830-1998 for Software Requirements Specification. The following conventions are used:

- "Shall" indicates a mandatory requirement
- "Should" indicates a recommended but not mandatory requirement
- "May" indicates an optional requirement

Priority levels:

- High: Essential features for initial release
- Medium: Important features that could be implemented in later iterations
- Low: Features that are desirable but not critical

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Project managers overseeing the development process
- Software developers implementing the system
- Quality assurance engineers testing the system

- Technical documentation writers
- Stakeholders evaluating the features and capabilities

For project managers and stakeholders, sections 1 and 2 provide a high-level overview. Developers should focus on sections 3 and 4, while QA engineers should pay special attention to sections 3, 4, and 5.

1.4 Product Scope

ResumeRover is a comprehensive HR management platform designed to revolutionize the traditional recruitment process through advanced artificial intelligence, natural language processing, blockchain verification, and real-time analytics. The system aims to:

- Transform manual resume screening into an automated, efficient process
- Provide deep analysis of resume content beyond simple keyword matching
- Ensure fair and unbiased candidate evaluation
- Verify candidate credentials through secure blockchain technology
- Offer actionable insights through comprehensive analytics dashboards
- Streamline the entire recruitment workflow from job posting to candidate selection

The system will benefit organizations by reducing time-to-hire, improving the quality of hires, minimizing bias in the recruitment process, and providing data-driven recruitment insights.

1.5 References

1. IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications
2. General Data Protection Regulation (GDPR)
3. Project Charter for ResumeRover
4. Internal Technical Standards Document

1.6 Requirement Engineering Process

The Requirements Engineering (RE) process followed these steps:

1. Requirements Elicitation

- Conducted internal brainstorming sessions and team discussions to gather feature ideas.
- Reviewed existing resume screening platforms and academic project guidelines to extract common functionalities and constraints.
- Analyzed similar projects and datasets to identify feature relevance for the ML models.

2. Requirements Analysis

- Classified requirements into functional, non-functional, and domain-specific categories.
- Prioritized features considering project objectives and user scenarios defined by the team.

3. Requirements Specification

- Documented requirements following IEEE SRS formatting and conventions.
- Assigned unique IDs, priorities, and descriptions to each requirement for clarity and traceability.

4. Requirements Validation

- Conducted peer reviews within the group and aligned features with interdependent groups (Groups 9–12) to avoid conflicts.

5. Requirements Management

- Used GitHub Projects integrated with GitHub Issues to track the status and progress of requirements.
- Maintained issue labels and backlog grooming as the scope evolved.

Tools Used: GitHub Issues + GitHub Projects (for backlog tracking), Google Docs (for collaborative drafting)

2. Overall Description

2.1 Domain Description

This system is developed in the domain of automated recruitment and intelligent resume screening. Traditional resume filtering is time-consuming, biased, and error-prone. The system aims to assist HR teams in evaluating applicants more fairly and efficiently using AI models, resume parsing, and data-driven scoring.

Stakeholders:

- System Administrators
- HR Managers
- Applicants

2.2 Product Perspective

ResumeRover is a new, standalone system that integrates multiple advanced technologies to create a comprehensive recruitment solution. The system consists of four major subsystems:

1. Resume Parsing & NLP Processing Subsystem
2. AI Candidate Ranking Subsystem
3. Blockchain-Based Verification Subsystem
4. Analytics & Dashboard Subsystem

While ResumeRover functions independently, it is designed to integrate with existing HR information systems and applicant tracking systems through APIs.

2.3 Product Functions

ResumeRover shall provide the following key functions:

- User account management with role-based access control
- Organization profile management with department hierarchies
- Detailed job role creation and requirement specification
- Resume intake and document processing
- Advanced natural language processing for resume content analysis
- AI-driven candidate ranking based on job requirements
- Blockchain-based verification of candidate credentials
- Real-time analytics and reporting dashboards
- Applicant profile and CV management

2.4 User Classes and Characteristics

HR Managers

- Primary users responsible for creating job roles, reviewing candidates, and making hiring decisions
- Need comprehensive views of candidate data and comparison tools
- Require administrative capabilities for managing team access and organizational settings
- Typically have moderate technical proficiency
- Use the system frequently during active recruitment campaigns

System Administrators

- Technical users responsible for system configuration and maintenance
- Require full access to system settings and administration tools
- High technical proficiency
- Occasional use focused on setup and troubleshooting

Job Applicants

- External users who interact with limited portions of the system
- Need simple, intuitive interfaces for CV submission and tracking
- Varying levels of technical proficiency
- One-time or occasional use
- May use various devices, including mobile phones

2.5 Operating Environment

- The system shall be deployed as a web-based application accessible through standard web browsers
- The system shall support the latest versions of Chrome, Firefox, Safari, and Edge browsers
- The system shall be responsive and function on desktop, tablet, and mobile devices
- The server infrastructure shall be cloud-based for scalability and reliability
- The system shall integrate with blockchain networks for verification features
- Database systems shall support high-volume data processing and analysis

- The system shall operate continuously with minimal planned downtime

2.6 Design and Implementation Constraints

- The system shall implement a microservices architecture to allow independent development and scaling of subsystems
- The blockchain verification component shall utilize Ethereum-compatible networks
- NLP processing shall leverage established frameworks like SpaCy and BERT models
- The system shall comply with data protection regulations including GDPR
- The system shall be designed to accommodate future machine learning model updates without requiring complete redeployment
- Authentication mechanisms shall support SSO integration
- Development practices shall adhere to Agile methodologies

2.7 User Documentation

The following user documentation shall be provided:

- System administration guide
- HR manager user manual
- Recruitment team quick start guide
- Applicant help resources
- API documentation for system integrations
- Blockchain verification process guide
- Video tutorials for key workflows
- Interactive help system within the application

2.8 Assumptions and Dependencies

- Users have reliable internet connectivity
- HR managers have basic knowledge of recruitment processes and terminology
- The system assumes the availability of blockchain networks for verification
- Integration with external systems depends on those systems having compatible APIs
- Performance depends on adequate cloud infrastructure resources
- Document processing accuracy depends on the quality of submitted documents

3. System Features

3.1 Account & Organization Management

3.1.1 User Registration and Authentication

- **Description**: The system shall provide secure user registration, authentication, and authorization mechanisms.
- **Functional Requirements**:
    - REQ-3.1.1.1: The system shall support user registration with email verification.

    - REQ-3.1.1.2: The system shall implement secure login with multi-factor authentication options.
    - REQ-3.1.1.3: The system shall support Single Sign-On (SSO) integration with common identity providers.
    - REQ-3.1.1.4: The system shall implement password policies including complexity requirements and expiration.
    - REQ-3.1.1.5: The system shall maintain an audit trail of authentication activities.
- **Priority**: High

3.1.2 Organization Profile Management

- **Description**: The system shall allow the creation and management of organizational profiles with department structures.
- **Functional Requirements**:
    - REQ-3.1.2.1: The system shall enable creation of organization profiles with basic information (name, industry, location).
    - REQ-3.1.2.2: The system shall support definition of department hierarchies within organizations.
    - REQ-3.1.2.3: The system shall allow assignment of users to specific departments.
    - REQ-3.1.2.4: The system shall support organization-specific branding options.
    - REQ-3.1.2.5: The system shall maintain version history of organization profile changes.
- **Priority**: High

3.1.3 Role-Based Access Control

- **Description**: The system shall implement role-based access control to govern user permissions.
- **Functional Requirements**:
    - REQ-3.1.3.1: The system shall support predefined roles (Admin, HR Manager, Recruiter, Verification Specialist).
    - REQ-3.1.3.2: The system shall allow administrators to create custom roles with specific permissions.
    - REQ-3.1.3.3: The system shall enable assignment of users to one or more roles.
    - REQ-3.1.3.4: The system shall restrict access to features and data based on user roles.
    - REQ-3.1.3.5: The system shall maintain an audit log of permission changes.
- **Priority**: High

3.1.4 White-Labeling and Branding

- **Description**: The system shall provide white-labeling capabilities for enterprise customers.
- **Functional Requirements**:

    - REQ-3.1.4.1: The system shall allow customization of colors and themes to match company branding.
    - REQ-3.1.4.2: The system shall support custom logo placement throughout the interface.
    - REQ-3.1.4.3: The system shall enable customization of email templates with company branding.
    - REQ-3.1.4.4: The system shall support custom domain configuration for application URLs.
    - REQ-3.1.4.5: The system shall provide branded candidate portals for resume submission.
- **Priority**: Medium

3.2 Job Role Creation

3.2.1 Job Specification Wizard

- **Description**: The system shall provide an interactive wizard for creating detailed job specifications.
- **Functional Requirements**:
    - REQ-3.2.1.1: The system shall guide users through a step-by-step process for defining job requirements.
    - REQ-3.2.1.2: The system shall capture basic job information (title, department, location, employment type).
    - REQ-3.2.1.3: The system shall allow specification of required skills.
    - REQ-3.2.1.4: The system shall enable definition of experience requirements.
    - REQ-3.2.1.5: The system shall support education requirements specification.
- **Priority**: High

3.3 Resume Parsing & NLP Processing

3.3.1 Document Intake System

- **Description**: The system shall provide a robust document intake system for resume collection.
- **Functional Requirements**:
    - REQ-3.3.1.1: The system shall support uploading of resumes in multiple formats (PDF, DOCX, TXT).
    - REQ-3.3.1.2: The system shall implement file validation and security scanning.
    - REQ-3.3.1.3: The system shall extract text content from various document formats.
    - REQ-3.3.1.4: The system shall process documents in batch or individually.
    - REQ-3.3.1.5: The system shall store original documents securely for reference.
    - REQ-3.3.1.6: The system shall handle documents with various layouts and formatting.
- **Priority**: High

3.3.2 NLP Engine

- **Description**: The system shall implement advanced NLP capabilities for resume analysis.
- **Functional Requirements**:
  - REQ-3.3.2.1: The system shall use SpaCy for structural analysis of resume content.
  - REQ-3.3.2.2: The system shall implement BERT models for semantic understanding.
  - REQ-3.3.2.3: The system shall identify and categorize resume sections (education, experience, skills).
  - REQ-3.3.2.4: The system shall extract named entities (companies, institutions, dates, locations).
  - REQ-3.3.2.5: The system shall analyze contextual information to determine relevance.
  - REQ-3.3.2.6: The system shall handle variations in terminology and phrasing.
- **Priority**: High

3.3.3 Entity Extraction Service

- **Description**: The system shall extract structured information from resume content.
- **Functional Requirements**:
  - REQ-3.3.3.1: The system shall extract contact information (name, email, phone, location).
  - REQ-3.3.3.2: The system shall extract education history with degree information and dates.
  - REQ-3.3.3.3: The system shall extract work experience with company names, titles, dates, and responsibilities.
  - REQ-3.3.3.4: The system shall extract skills and technologies with context.
  - REQ-3.3.3.5: The system shall extract certifications and qualifications with dates.
  - REQ-3.3.3.6: The system shall extract projects and accomplishments with metrics where available.
- **Priority**: High

3.3.4 Contextual Analysis Module

- **Description**: The system shall analyze the relevance of resume content to job requirements.
- **Functional Requirements**:
  - REQ-3.3.4.1: The system shall map extracted resume information to job requirements.
  - REQ-3.3.4.2: The system shall evaluate the relevance of experience to job responsibilities.
  - REQ-3.3.4.3: The system shall assess skill alignment with job requirements.
  - REQ-3.3.4.4: The system shall determine the recency and duration of relevant experience.

    - REQ-3.3.4.5: The system shall identify transferable skills not explicitly mentioned in requirements.
- **Priority**: High

3.3.5 Data Standardization Service

- **Description**: The system shall normalize extracted information into a consistent format.
- **Functional Requirements**:
    - REQ-3.3.5.1: The system shall standardize job titles across industries.
    - REQ-3.3.5.2: The system shall normalize company and institution names.
    - REQ-3.3.5.3: The system shall standardize skill terminology and technology names.
    - REQ-3.3.5.4: The system shall normalize educational qualifications and degree names.
    - REQ-3.3.5.5: The system shall provide confidence scores for standardization accuracy.
- **Priority**: Medium

3.4 AI Candidate Ranking

3.4.1 Scoring Engine

- **Description**: The system shall calculate candidate match scores based on extracted information.
- **Functional Requirements**:
    - REQ-3.4.1.1: The system shall evaluate candidates using a decision tree model.
    - REQ-3.4.1.2: The system shall calculate overall match percentage scores.
- **Priority**: High

3.4.2 Decision Tree Models

- **Description**: The system shall use decision tree models for complex evaluation of candidates.
- **Functional Requirements**:
    - REQ-3.4.2.1: The system shall implement ensemble models for robust candidate evaluation.
    - REQ-3.4.2.2: The system shall handle both quantitative and qualitative factors in ranking.
    - REQ-3.4.2.3: The system shall adapt evaluation based on job role and industry.
- **Priority**: High

3.4.3 Fairness Module

- **Description**: The system shall implement fairness measures to prevent bias in candidate evaluation.
- **Functional Requirements**:

    - REQ-3.4.3.1: The system shall integrate IBM AI Fairness 360 toolkit for bias detection.
    - REQ-3.4.3.2: The system shall identify potential bias in ranking across demographic groups.
    - REQ-3.4.3.3: The system shall apply mitigation techniques when bias is detected.
    - REQ-3.4.3.4: The system shall provide transparency in fairness measures.
    - REQ-3.4.3.5: The system shall generate fairness reports for compliance purposes.
- **Priority**: High

3.4.4 Ranking Optimization Service

- **Description**: The system shall continuously improve ranking algorithms based on feedback.
- **Functional Requirements**:
    - REQ-3.4.4.1: The system shall track hiring outcomes to evaluate ranking effectiveness.
    - REQ-3.4.4.2: The system shall incorporate explicit feedback from HR users.
    - REQ-3.4.4.3: The system shall implement A/B testing for ranking algorithm improvements.
    - REQ-3.4.4.4: The system shall maintain model versioning and performance tracking.
    - REQ-3.4.4.5: The system shall provide performance metrics for ranking accuracy.
- **Priority**: Medium

3.4.5 Explanation Generator

- **Description**: The system shall provide transparent reasoning for ranking decisions.
- **Functional Requirements**:
    - REQ-3.4.5.1: The system shall generate natural language explanations for match scores.
    - REQ-3.4.5.2: The system shall highlight key factors influencing ranking decisions.
    - REQ-3.4.5.3: The system shall provide detailed breakdowns of scores by category.
    - REQ-3.4.5.4: The system shall visualize strengths and weaknesses in candidate profiles.
    - REQ-3.4.5.5: The system shall link explanations to specific job requirements.
- **Priority**: Medium

3.5 Blockchain-Based Verification

3.5.1 Smart Contract Layer

- **Description**: The system shall implement blockchain-based contracts for credential verification.
- **Functional Requirements**:

    - REQ-3.5.1.1: The system shall develop Ethereum-based smart contracts for credential verification.
    - REQ-3.5.1.2: The system shall implement secure hashing of credential documents.
    - REQ-3.5.1.3: The system shall store verification status on the blockchain.
    - REQ-3.5.1.4: The system shall provide access control mechanisms for verification data.
    - REQ-3.5.1.5: The system shall implement upgradeability patterns for contract improvements.
- **Priority**: High

3.5.2 Oracle Integration

- **Description**: The system shall connect with external data sources for verification.
- **Functional Requirements**:
    - REQ-3.5.2.1: The system shall integrate Chainlink oracles for external data verification.
    - REQ-3.5.2.2: The system shall implement connections to educational institution databases.
    - REQ-3.5.2.3: The system shall create interfaces with professional certification authorities.
    - REQ-3.5.2.4: The system shall develop fallback mechanisms for oracle failures.
    - REQ-3.5.2.5: The system shall aggregate results from multiple verification sources.
- **Priority**: Medium

3.5.3 Verification Status Tracker

- **Description**: The system shall monitor and report verification progress.
- **Functional Requirements**:
    - REQ-3.5.4.1: The system shall track verification status of individual credentials.
    - REQ-3.5.4.2: The system shall provide status updates to HR users and applicants.
    - REQ-3.5.4.3: The system shall implement notification mechanisms for status changes.
    - REQ-3.5.4.4: The system shall manage verification request prioritization.
    - REQ-3.5.4.5: The system shall provide retry mechanisms for failed verifications.
- **Priority**: High

3.5.4 Immutable Record Manager

- **Description**: The system shall maintain permanent verification history.
- **Functional Requirements**:
    - REQ-3.5.5.1: The system shall create immutable records of completed verifications.

    - REQ-3.5.5.2: The system shall generate verification certificates with blockchain references.
    - REQ-3.5.5.3: The system shall provide audit trails for verification activities.
    - REQ-3.5.5.4: The system shall implement secure storage of verification history.
    - REQ-3.5.5.5: The system shall enable verification record retrieval through public keys.
- **Priority**: Medium

3.6 Analytics & Dashboard

3.6.1 Data Aggregation Service

- **Description**: The system shall collect and process metrics from all subsystems.
- **Functional Requirements**:
    - REQ-3.6.1.1: The system shall aggregate data from parsing, ranking, and verification subsystems.
    - REQ-3.6.1.2: The system shall implement time-series data storage for temporal analysis.
    - REQ-3.6.1.3: The system shall process real-time events from recruitment activities.
- **Priority**: Medium

3.6.2 Visualization Engine

- **Description**: The system shall generate interactive charts and graphs for data visualization.
- **Functional Requirements**:
    - REQ-3.6.2.1: The system shall implement Plotly-based visualization components.
    - REQ-3.6.2.2: The system shall provide interactive dashboard layouts.
    - REQ-3.6.2.3: The system shall generate various chart types (line, bar, pie, scatter, etc.).
    - REQ-3.6.2.4: The system shall support filtering and time range selection.
    - REQ-3.6.2.5: The system shall enable drill-down capabilities for detailed analysis.
- **Priority**: Medium

3.6.3 Predictive Analytics Module

- **Description**: The system shall implement machine learning models for forecasting recruitment metrics.
- **Functional Requirements**:
    - REQ-3.6.3.1: The system shall develop ARIMA models for application volume forecasting for a specific job role.
    - REQ-3.6.3.2: The system shall track prediction accuracy and improve models over time.
- **Priority**: Low

3.6.4 Real-time Monitoring Service

- **Description**: The system shall provide live updates on recruitment status.
- **Functional Requirements**:
  - REQ-3.6.5.1: The system shall display real-time application statistics.
  - REQ-3.6.5.2: The system shall implement threshold-based alerts for key metrics.
  - REQ-3.6.5.3: The system shall deliver notifications through multiple channels.
- **Priority**: Medium

3.7 CV Management

3.7.1 CV Storage System

- **Description**: The system shall provide secure document storage for applicant CVs.
- **Functional Requirements**:
  - REQ-3.7.1.1: The system shall securely store CVs for each applicant.
  - REQ-3.7.1.2: The system shall implement versioning for CV revisions.
  - REQ-3.7.1.3: The system shall enable metadata tagging for CV organization.
  - REQ-3.7.1.4: The system shall categorize CVs by job type or industry.
  - REQ-3.7.1.5: The system shall provide CV comparison tools.
- **Priority**: High

3.8 Recruitment Campaign Management

3.8.1 Campaign Creation and Tracking

- **Functional Requirements**:
  - REQ-3.8.1.3: The system shall provide customizable landing pages for applications.
  - REQ-3.8.1.4: The system shall track application volume and source metrics.
  - REQ-3.8.1.5: The system shall implement application deadline management.
  - REQ-3.8.1.6: The system shall provide controls to pause or close recruitment campaigns.
- **Priority**: High

3.8.2 Candidate Communications

- **Description**: The system shall support candidate communication by automating email notifications based on application status. Candidate email addresses are extracted during resume parsing and stored in the database. The system will use this information to facilitate communication both automatically and through HR actions.
- **Functional Requirements**:
  - REQ-3.8.2.1: The system shall automatically send rejection emails to candidates who are not shortlisted, using the extracted email addresses.
  - REQ-3.8.2.2: The system shall provide HR personnel with an interface to review shortlisted candidates and either accept or reject them.

    - REQ-3.8.2.3: The system shall send acceptance or rejection emails to shortlisted candidates based on the HR's action (i.e., pressing “Accept” or “Reject” buttons).
    - REQ-3.8.2.4: The system shall store the status of each candidate's communication in the database (e.g., “Rejected – Email Sent”, “Shortlisted – Awaiting HR Decision”).
- **Priority**: Medium

3.8.3 Candidate Evaluation Dashboard

- **Description**: The system shall provide comprehensive tools for evaluating candidates.
- **Functional Requirements**:
    - REQ-3.8.3.1: The system shall display ranked candidate lists with match percentages.
    - REQ-3.8.3.2: The system shall display verification status indicators for credentials.
- **Priority**: High

4. External Interface Requirements

4.1 User Interfaces

4.1.1 HR Manager Interface

- The HR manager interface shall provide a comprehensive dashboard with access to all system functions.
- The interface shall be responsive and work on desktop and tablet devices.
- The interface shall implement modern UI/UX best practices with intuitive navigation.
- The interface shall include customizable views for different recruitment activities.
- The interface shall provide quick access to active campaigns and candidate lists.

4.1.2 Applicant Interface

- The applicant interface shall be clean, simple, and focused on CV submission and tracking.
- The interface shall be fully responsive across desktop, tablet, and mobile devices.
- The interface shall minimize clicks required for CV submission.
- The interface shall clearly communicate application status and next steps.
- The interface shall support accessibility standards for users with disabilities.

4.1.3 Administrator Interface

- The administrator interface shall provide comprehensive system configuration options.
- The interface shall include user management and permission controls.
- The interface shall display system health and performance metrics.
- The interface shall provide tools for data management and backup.

- The interface shall include configuration options for all system components.

4.2 Hardware Interfaces

- The system shall be compatible with standard computing hardware.
- The system shall operate on devices meeting minimum browser requirements.
- The system does not require special hardware components for operation.
- The system shall be accessible from mobile devices for specific functions.
- The system shall operate regardless of the user's input peripherals.

4.3 Software Interfaces

4.3.1 Browser Compatibility

- The system shall support the latest versions of Chrome, Firefox, Safari, and Edge browsers.
- The system shall provide graceful degradation for older browser versions.
- The system shall implement responsive design practices for various screen sizes.
- The system shall notify users of incompatible browser environments.
- The system shall maintain consistent functionality across supported browsers.

4.3.2 Integration Interfaces

- The system shall provide REST APIs for integration with external systems.
- The system shall implement webhook capabilities for event-driven integrations.
- The system shall support standard authentication mechanisms for API security.
- The system shall provide detailed API documentation with examples.
- The system shall implement rate limiting and throttling for API stability.

4.3.3 Blockchain Network Interfaces

- The system shall connect to Ethereum-compatible blockchain networks.
- The system shall implement Web3.js/Web3.py for blockchain interaction.
- The system shall support both test networks and production networks.
- The system shall handle network disruptions gracefully with retry mechanisms.
- The system shall implement secure key management for blockchain transactions.

4.4 Communications Interfaces

4.4.1 Email Communication

- The system shall send notifications and updates via email.
- The system shall implement email templates with dynamic content.

4.4.2 API Communication

- The system shall communicate with external systems via secure HTTPS.
- The system shall implement standard authentication mechanisms for API calls.
- The system shall handle timeouts and connection failures gracefully.

- The system shall log all external communication for troubleshooting.
- The system shall implement retry logic for failed communications.

5. Non-Functional Requirements

5.1 Performance Requirements

5.1.1 Response Time

- The system shall process a single resume within 30 seconds.
- The system shall provide search results within 2 seconds.
- The dashboard shall load within 3 seconds on standard connections.
- The system shall handle batch processing of up to 100 resumes concurrently.
- User interface interactions shall have a response time under 1 second.

5.1.2 Throughput

- The system shall support processing of at least 1,000 resumes per day.
- The system shall handle at least 100 concurrent users without performance degradation.
- The system shall support at least 50 concurrent resume uploads.
- The API shall handle at least 100 requests per minute.
- The system shall process at least 20 verification requests concurrently.

5.1.3 Capacity

- The system shall store at least 1 million resumes with associated metadata.
- The system shall support at least 1,000 organizations with separate data domains.
- The system shall handle at least 10,000 job roles across all organizations.
- The system shall maintain verification records for at least 5 years.
- The analytics subsystem shall store at least 3 years of historical data.

5.1.4 Scalability

- The system architecture shall support horizontal scaling of all components.
- The system shall maintain performance levels as data volume increases.
- The system shall implement caching mechanisms for frequently accessed data.
- The system shall support cloud-based auto-scaling based on load.
- The database design shall optimize for growing datasets.
- The AI model shall be adaptable to various job roles and industries without requiring major redesigns.

5.2 Safety Requirements

- The system shall implement data validation to prevent data corruption.
- The system shall provide backup and recovery mechanisms for all critical data.
- The system shall implement transaction management to maintain data consistency.
- The system shall provide error handling to prevent system crashes.

- The system shall log all critical operations for troubleshooting and recovery.

5.3 Security Requirements

5.3.1 Authentication and Authorization

- The system shall enforce strong password policies.
- The system shall implement multi-factor authentication options.
- The system shall use role-based access control for all functions.
- The system shall maintain detailed access logs.
- The system shall automatically log out inactive sessions after 30 minutes.

5.3.2 Data Protection

- The system shall encrypt all sensitive data at rest using AES-256.
- The system shall use TLS 1.3 for all data in transit.
- The system shall implement secure coding practices to prevent common vulnerabilities.
- The system shall regularly scan for security vulnerabilities.
- The system shall anonymize data used for analytics when possible.

5.3.3 Compliance

- The system shall comply with GDPR requirements for EU data subjects.
- The system shall implement data retention policies according to regulations.
- The system shall provide mechanisms for data subject access requests.
- The system shall support data portability requirements.
- The system shall maintain audit trails for compliance verification.

5.4 Software Quality Attributes

5.4.1 Reliability

- The system shall have an uptime of at least 99.9% (excluding planned maintenance).
- The system shall implement automated error recovery mechanisms.
- The system shall handle unexpected inputs gracefully without crashing.
- The system shall maintain data integrity during unexpected shutdowns.
- The system shall provide automated backup mechanisms.

5.4.2 Availability

- The system shall be available 24/7 except for scheduled maintenance windows.
- Scheduled maintenance shall be performed during low-usage periods.
- The system shall notify users of planned downtime at least 48 hours in advance.
- The system shall implement redundancy for critical components.
- The system shall recover from failures within 15 minutes.

5.4.3 Maintainability

- The system shall follow a microservices architecture for independent component updates.
- The system shall implement comprehensive logging for troubleshooting.
- The system shall use containerization for consistent deployment environments.
- The system shall provide administrative tools for system maintenance.
- The system code shall follow established coding standards and practices.

5.4.4 Portability

- The system shall be deployable on major cloud platforms (AWS, Azure, GCP).
- The system shall use containerization to ensure environment consistency.
- The system shall minimize platform-specific dependencies.
- The system shall support database migration between supported platforms.
- The system shall document all external dependencies.

5.4.5 Usability

- The system shall provide intuitive navigation with minimal learning curve.
- The system shall implement consistent UI patterns across all interfaces.
- The system shall provide contextual help and tooltips for complex features.
- The system shall support keyboard shortcuts for common operations.
- The system shall provide clear error messages with suggested actions.

5.4.6 Accuracy

- The system shall maintain a prediction accuracy such that the Mean Absolute Error (MAE) does not exceed 2 units on the validation set.
- The system shall maintain an R^2 score of at least 0.1 on the test dataset to ensure reliable resume scoring performance.

5.5 Business Rules

- The system shall comply with equal opportunity employment regulations.
- The system shall implement bias detection and mitigation in candidate ranking.
- The system shall maintain separation of duties for verification processes.
- The system shall enforce data retention policies according to legal requirements.
- The system shall support organizational approval workflows for recruitment decisions.

6. Other Requirements

6.1 Database Requirements

- The system shall implement a scalable database architecture.
- The system shall support both SQL and NoSQL databases for different data types.
- The system shall implement database sharding for large-scale deployments.
- The system shall provide database backup and recovery mechanisms.
- The system shall optimize queries for performance with large datasets.

6.2 Internationalization Requirements

- The system shall support multiple languages in the user interface.
- The system shall handle international date and time formats.
- The system shall support international character sets and encodings.
- The system shall accommodate different name formats across cultures.
- The system shall provide region-specific templates and configurations.

6.3 Legal Requirements

- The system shall comply with data protection regulations in operating jurisdictions.
- The system shall implement features required for GDPR compliance.
- The system shall provide mechanisms for fulfilling data subject rights.
- The system shall maintain appropriate records for legal compliance.
- The system shall implement appropriate data processing agreements.

Appendix A: Glossary

ATS: Applicant Tracking System - Software that manages the recruitment process.

BERT: Bidirectional Encoder Representations from Transformers - A natural language processing model.

CV: Curriculum Vitae - A document detailing a person's education, professional experience, skills, and achievements.

GDPR: General Data Protection Regulation - EU regulation on data protection and privacy.

HR: Human Resources - The department responsible for recruiting, hiring, and managing employees.

NLP: Natural Language Processing - A field of AI focused on enabling computers to understand human language.

Oracle: In blockchain, a mechanism that provides external data to a blockchain.

Smart Contract: Self-executing contracts with the terms directly written into code.

SpaCy: An open-source software library for advanced natural language processing.

SSO: Single Sign-On - An authentication method that allows users to access multiple applications with one set of credentials.

Appendix B: Analysis Models

B.1 System Context Diagram

[Description of system context diagram showing ResumeRover and its interactions with external entities]

B.2 Data Flow Diagram

[Description of data flow diagram showing the movement of data through the system]

B.3 Entity Relationship Diagram

[Description of entity relationship diagram showing the database schema]

B.4 State Transition Diagram

[Description of state transition diagram showing the states of a resume in the system]

Appendix C: Issues List

1. The integration with specific external verification sources needs further investigation.
2. Performance benchmarks for NLP processing of large resume volumes require validation.
3. Legal requirements for credential verification across different jurisdictions need clarification.
4. Optimal blockchain implementation strategy needs a final decision (private vs. public chain).
5. Detailed compliance requirements for specific industries need further research.
    ○