**Software Requirements Specification (SRS)**

for  
**SkillSync**  
Version 1.0  
Prepared by Proconnect  
September 2025

**Table of Contents**

1. Introduction  
   1.1 Purpose  
   1.2 Document Conventions  
   1.3 Intended Audience and Reading Suggestions  
   1.4 Product Scope  
   1.5 References
2. Overall Description  
   2.1 Product Perspective  
   2.2 Product Functions  
   2.3 User Classes and Characteristics  
   2.4 Operating Environment  
   2.5 Design and Implementation Constraints  
   2.6 User Documentation  
   2.7 Assumptions and Dependencies
3. External Interface Requirements  
   3.1 User Interfaces  
   3.2 Hardware Interfaces  
   3.3 Software Interfaces  
   3.4 Communications Interfaces
4. System Features  
   4.1 Resume Parsing and Skill Extraction  
   4.2 Quiz Generation and Skill Verification  
   4.3 Messaging and Notification  
   4.4 Hiring Workflow  
   4.5 Skill Status Management
5. Other Nonfunctional Requirements  
   5.1 Performance Requirements  
   5.2 Safety Requirements  
   5.3 Security Requirements  
   5.4 Software Quality Attributes
6. Glossary

**1. Introduction**

**1.1 Purpose**

This document describes the requirements for the development of the **SkillSync** platform. The system automates skill extraction from user-uploaded resumes, verifies extracted skills through dynamic quizzes, enables direct messaging between candidates and employers, and streamlines skill-based hiring. This document is intended for developers, testers, project stakeholders, and potential contributors.Sample-SRS-Document.pdf

**1.2 Document Conventions**

* This document follows IEEE SRS guidelines.
* Definitions, requirements, and models are clearly separated and systematically numbered.
* All figures and tables are referenced in relevant contexts.

**1.3 Intended Audience and Reading Suggestions**

* **Developers:** For implementation and system design guidance.
* **QA/Testers:** To validate the system functions against user stories and requirements.
* **Employers/HR Professionals:** For understanding system behavior, hiring workflows, and skill verification processes.
* **Students and Candidates:** For understanding profile management and skill verification.

**1.4 Product Scope**

SkillSync is a cloud-based web platform designed to digitize and verify skill portfolios of users via AI-driven resume parsing and quizzes. It facilitates transparent, skills-based hiring through real-time messaging and notifications, streamlining the candidate-employer interaction.

**1.5 References**

* IEEE SRS Template ([https://www.ieee.org](https://www.ieee.org/))
* NLP Libraries (spaCy, BERT, NLTK)
* RESTful API best practices

**2. Overall Description**

**2.1 Product Perspective**

SkillSync is a modular system designed for extensibility and smooth integration with third-party job boards, HRMS, and learning systems. AI-powered microservices are leveraged for document processing, quiz management, and messaging.Sample-SRS-Document.pdf

**2.2 Product Functions**

* **Resume Parsing:** Extracts skills using NLP from PDF resumes.
* **Skill Assignment:** Skills are assigned as unverified until the candidate completes verification.
* **Quiz Engine:** Generates skill-based quizzes and records outcomes.
* **Messaging:** Enables communication between employers and candidates.
* **Hiring Workflow:** Allows employers to send direct, actionable hiring invitations.
* **Skill Status Management:** Tracks and updates skill verification in real time.

**2.3 User Classes and Characteristics**

* **Candidate:** Seeks employment, manages resume and verified skills, completes quizzes.
* **Employer/Recruiter:** Searches for talent, views candidates by skills, initiates hiring, and communicates directly.
* **Admin:** Monitors system usage, manages users and content, maintains the platform.

**2.4 Operating Environment**

* Web-based (browser support: Chrome, Firefox, Edge, Safari)
* Supports desktop and mobile devices
* Hosted on cloud infrastructure (e.g., AWS, Azure)
* Backend in Python or Node.js, frontend in React or Angular

**2.5 Design and Implementation Constraints**

* User authentication via OAuth 2.0 or similar secure protocol
* All personal data encrypted in transit and at rest
* The system must support at least 10,000+ concurrent users
* Inter-service communication via RESTful APIs

**2.6 User Documentation**

* User manual, FAQs, and workflow diagrams provided in the Help section
* Video tutorials for candidates and employers

**2.7 Assumptions and Dependencies**

* Users have basic computer literacy
* Candidates can upload PDF documents
* Quiz bank is curated by domain experts or generated by AI

**3. External Interface Requirements**

**3.1 User Interfaces**

* **Candidate Dashboard:** Profile, resume upload, skill status (verified/unverified), quiz access, messages.
* **Quiz Interface:** Personalized quiz interface with timer and immediate feedback.
* **Employer Dashboard:** Talent search by skill, candidate profiles, direct messaging, hiring actions.
* **Notification Center:** Real-time job, message, and assessment updates.
* **Admin Console:** User management, analytics, platform configuration.

**3.2 Hardware Interfaces**

* Standard PC/laptop or smartphone with at least 4GB RAM and modern web browser
* No specialized hardware required

**3.3 Software Interfaces**

* PDF parsing and NLP microservices for resume ingestion
* Question bank/AI quiz generator microservice
* Messaging service (WebSockets/REST)
* Email/SMS gateway for notifications
* Integration APIs for external HRMS/job boards

**3.4 Communications Interfaces**

* Secure HTTPS for all client-server transactions
* WebSockets for real-time messaging and notifications

**4. System Features**

**4.1 Resume Parsing and Skill Extraction**

* **Description:** Candidates upload a PDF resume; the system parses content using an NLP pipeline to extract relevant skills.
* **Functional Requirements:**
  + System supports .pdf uploads only.
  + Extracted skills are auto-tagged as "unverified".
  + Candidates can edit, remove, or request manual review of extracted skills.

**4.2 Quiz Generation and Skill Verification**

* **Description:** For every unverified skill, the platform auto-generates a domain-specific quiz. On passing, the skill is marked as verified.
* **Functional Requirements:**
  + Quiz questions sourced from AI or a domain-specific bank.
  + Instant feedback after completion.
  + Admin/candidate can request re-quiz for failed skills.

**4.3 Messaging and Notification**

* **Description:** Candidates and employers can communicate directly through secure messaging.
* **Functional Requirements:**
  + Real-time WebSockets/chat.
  + Message history visible to both parties.
  + Email/SMS/app notification on hiring actions.

**4.4 Hiring Workflow**

* **Description:** Employers shortlist verified candidates, initiate hiring with an instant notification/offering. Candidates act with "Accept" or "No".
* **Functional Requirements:**
  + Employer can send direct hiring request.
  + Candidate receives notification, responds within system.
  + System records and timestamps hiring decisions.

**4.5 Skill Status Management**

* **Description:** System tracks each skill’s status from extraction to verification, including quiz performance.
* **Functional Requirements:**
  + Skill status updates in real time upon verification.
  + Employers see only verified skills when searching for candidates.
  + All status changes are logged for audit.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* Resume parsing and skill extraction must complete within 10 seconds per resume.
* Quizzes load in under 3 seconds.
* Messaging has under 1 second latency.
* 99.95% uptime; robust cloud failover and autoscaling support.

**5.2 Safety Requirements**

* Periodic backups; disaster recovery processes.
* Graceful handling of unexpected system failures; unsaved candidate progress preserved.

**5.3 Security Requirements**

* All sensitive data (resumes, messages, assessments) encrypted at rest and in transit.
* Mandatory user authentication and authorization.
* GDPR-compliant user data management (deletion, export, consent).

**5.4 Software Quality Attributes**

* **Usability:** Clean, intuitive UI for both candidates and employers.
* **Reliability:** Fault-tolerant microservices; rigorous monitoring.
* **Maintainability:** Modular, tested codebase; clear separation of API and business logic.
* **Scalability:** Containerized deployment supports horizontal scaling.

**6. Glossary**

* **Candidate:** User seeking employment, uploads resume, completes quizzes.
* **Employer:** User seeking to hire candidates, can message, initiate interview/hiring.
* **NLP (Natural Language Processing):** AI technology used for skill extraction from resumes.
* **Skill (Unverified/Verified):** Tag applied to each extracted or proven candidate ability.
* **Quiz:** Assessment engine to verify skill claims via targeted questions.
* **Direct Hire:** Employer initiates one-click hiring; candidate responds instantly.
* **Microservice:** Independently deployable component responsible for a specific system function.

This SRS document establishes the detailed expectation and modular structure for the envisioned skill-centric hiring platform, following modern SaaS and AI-driven standards and best practices.Sample-SRS-Document.pdf

1. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/72970906/8b1fc4af-2054-4bc6-afc6-048dd134d77a/Sample-SRS-Document.pdf>

The best model architecture for this kind of resume-driven, skill-verification, and hiring platform is a **modular microservices architecture** powered by AI, with each core business feature implemented as an independent service and orchestrated through cloud-based APIs.

**Modular Microservices Architecture**

**Key Features:**

* **Resume Parsing Microservice:** Utilizes deep learning/NLP models (like transformers or spaCy) for extracting structured skills data from resumes. This service operates independently and communicates results through APIs.
* **Skill Verification/Assessment Microservice:** Maintains a quiz bank and dynamically generates skill quizzes, supporting rule-based and AI-assisted question selection for each unique skill profile.
* **User & Authentication Service:** Handles OAuth2.0 logins, user roles (candidate/employer/admin), and profile management.
* **Messaging & Notification Service:** Real-time messaging is managed by WebSockets or event-driven messaging architecture to enable instant communication for outreach, hiring, and quiz notifications.
* **Hiring Orchestration Service:** Manages hiring workflows, actions, and logging so that hiring requests, responses, and skill verifications are tightly integrated.
* **APIs & Integrations Layer:** Allows horizontal scaling, integration with third-party HRMS, ATS, or job boards, and ensures new features/plugins can be added without affecting the overall system.

**Deployment and DevOps:**

* **All services are containerized (Docker)** and deployed via orchestrators like Kubernetes for auto-scaling and resilience.
* **Event-driven communication** (Kafka/RabbitMQ) ensures loose coupling between modules and reliable notifications.
* **REST or GraphQL APIs** enable both scalable frontend/backend operations and easy extensibility.

**Why This Works Best**

* **Scalability:** Each feature (resume parsing, quiz engine, messaging, etc.) can scale separately as usage grows.
* **Maintainability:** Independent modules reduce fragility and simplify updates or technology changes.
* **AI Infusion:** Natural language models can be continuously updated/expanded in the parsing and quiz services without downtime for the rest of the platform.
* **Adaptability:** Easily extend for new assessments, employer features, or analytics by plugging in new services.

This architecture reflects industry best practices for modern AI-powered recruitment and skill management platforms, with proven effectiveness in similar solutions.