**1.Introduction to Project System**

The RecruitAI Screening Agent is an end-to-end automated platform designed to modernize the initial phases of talent acquisition. By leveraging Large Language Models (LLMs) and a proprietary data compression engine, the system transforms the traditional, labor-intensive resume review process into a high-speed, data-driven workflow.

In modern recruitment, HR teams often face an "application overload" where hundreds of candidates apply for a single role. Manual screening is not only slow but prone to human fatigue and unconscious bias.

The Recruit AI system is engineered to:

Identify Top Talent: Use semantic analysis to find candidates whose skills match the *intent* of a job description, not just specific keywords.

Reduce Operational Costs: Utilize a Compression Engine to minimize the number of tokens sent to LLMs, reducing API costs by up to $60\%$.

Standardize Evaluation: Ensure every candidate is measured against the same objective criteria, improving fairness and compliance.

The system is engineered to deliver three primary pillars of value:

Semantic Precision (Identifying Top Talent): Unlike traditional Applicant Tracking Systems (ATS) that rely on rigid, literal keyword matching, RecruitAI utilizes semantic analysis. It understands the *intent* behind a job description. For example, if a role requires "leadership experience," the agent can recognize a candidate’s background in "team orchestration" or "project oversight" as a match, even if the exact keywords are missing.

Architectural Efficiency (Reducing Costs): A common hurdle in AI adoption is the high cost of API tokens. Our proprietary Compression Engine strips away resume boilerplate and formatting noise before it reaches the LLM. This allows the system to reduce operational API costs by up to $60\%$, making enterprise-scale AI screening financially sustainable.

Evaluation Standardization: By measuring every candidate against an identical, objective set of criteria, the agent eliminates the variability of human mood or fatigue. This ensures a standardized evaluation process that significantly bolsters corporate fairness and compliance initiatives.

1. **System Overview**

The RecruitAI Screening Agent serves as an automated intermediary between raw candidate data and hiring decisions. Unlike standard LLM implementations, it utilizes a proprietary Compression Engine to reduce redundant data (boilerplate text, formatting noise) before analysis. This ensures faster processing times and significantly lower operational costs.

**Key Value Propositions**

Cost Efficiency: Real-time tracking of savings through token optimization.

High Precision: Maintain a high "Match Quality" (currently $89\%$) by focusing on core competencies.

Scalability: Process large batches of candidates without hitting standard LLM context window limits.

2. Technical Workflow

The following diagram illustrates how the RecruitAI agent processes data from ingestion to final reporting:

1. Ingestion: The system pulls data from the "Candidates" and "Job Descriptions" modules.
2. Compression Engine: The "secret sauce." It tokenizes resumes and strips away non-essential boilerplate to minimize the context window.
3. LLM Screening: The compressed data is matched against the job requirements using a high-fidelity LLM.
4. Reporting: Results are populated into the dashboard, updating the Efficiency Report and Match Quality metrics.

3. Dashboard Components

Based on the Screening Dashboard, the interface is divided into four primary functional zones:

### A. Performance Analytics (Top Row)

* Total Candidates: Tracks the volume of the talent pool and week-over-week growth.
* Time Saved: A productivity metric calculating the man-hours saved via automation.
* Cost Optimized: A financial tracker showing the ROI of the Compression Engine.
* Match Quality: A confidence score representing how well the AI’s selections align with job requirements.

1. **SYSTEM REQUIREMENT SPECIFICATION**

The Software Requirements Specification (SRS) serves as the definitive roadmap for the RecruitAI Screening Agent. Its primary goal is to provide a comprehensive description of the system’s behavior, performance expectations, and constraints.

By clearly documenting these requirements, we ensure that the development team, stakeholders, and end-users (recruiters) are perfectly aligned. This document minimizes "feature creep" and ensures that every line of code written contributes directly to the project's core objectives: speed, cost-efficiency, and matching precision.

**Functional Requirements (FR)**

Functional requirements define the specific actions the system must be able to perform. These are the "tasks" the user expects the agent to complete

2.3 Non-Functional Requirements (NFR)

*Non-functional requirements define the attributes of the system—how it should perform rather than what it does. These are the "qualities" of the software.*

🚀 Performance & Scalability

* Latency: The compression and screening of a single standard resume (2 pages) must be completed in under 5 seconds.
* Throughput: The system must support the concurrent screening of up to 500 candidates without service degradation.

🛡️ Security & Privacy

* Data Encryption: All candidate data must be encrypted at rest using AES-256 and in transit via TLS 1.3.
* PII Masking: The system should have an optional "Blind Hiring" mode to mask Names, Gender, and Locations during the initial AI screening to prevent bias.

**Hardware Requirements**

To ensure the "Match Quality" and "Compression" logic operates at peak performance ($<5s$ per resume), the following hardware specifications are recommended:

A. Server/Development Side

If the system is hosted on-premise or during the development phase, these specs are necessary to handle the tokenization and AI orchestration.

* Processor (CPU): Intel Core i7 or AMD Ryzen 7 (12th Gen or higher) for local development; Octa-core vCPU for cloud instances.
* Memory (RAM): Minimum 16 GB (32 GB recommended for handling concurrent batch screenings).
* Storage: 512 GB NVMe SSD (High-speed read/write is essential for parsing large PDF batches).
* GPU (Optional): NVIDIA RTX 3060 or higher with 8GB VRAM (Only required if running local open-source models like Llama-3; not required for API-based setups).

**Software Requirements**

The software stack is built for Agentic-Ready Architecture, prioritizing speed and high connectivity.

A. Backend & AI Stack

* Operating System: Linux (Ubuntu 22.04 LTS recommended) or macOS for development.
* Programming Language: Python 3.10+ (The industry standard for AI and LLM orchestration).
* Framework: FastAPI (Chosen for its asynchronous capabilities, allowing multiple candidates to be screened simultaneously).
* AI Orchestration: LangChain or LangGraph (To manage the "Compression → Screening → Scoring" agent workflow).
* LLM API: OpenAI GPT-4o or Google Gemini 1.5 Pro (Primary "brains" of the screening logic).

**System Users**

The RecruitAI Screening Agent is designed for a multi-tenant HR environment. While the AI performs the heavy lifting of data compression and initial ranking, the system is built to support three distinct tiers of human interaction:

1. **Administrative Users:** Responsible for the technical health and financial oversight of the platform.
2. **Operational Users (Recruiters):** The primary drivers of the system who manage the day-to-day candidate pipeline.
3. **Executive Users (Hiring Managers):** Strategic consumers of the data who use the generated shortlists to make final hiring decisions.

**Description of User Roles**

Each role is assigned a specific set of permissions within the dashboard to ensure "Least Privilege" security protocols.

1. System Administrator (Admin)

The Admin ensures the engine is running smoothly and cost-effectively.

* Key Responsibilities: Managing API integrations (OpenAI/Gemini keys), monitoring total credit usage (e.g., the 10,000 credit limit), and adding/removing team members.
* Access Level: Full CRUD (Create, Read, Update, Delete) permissions across all modules.
* Dashboard Focus: "Cost Optimized" and "Credits Used" metrics.

2. Talent Acquisition Specialist (Recruiter)

The power user. They are responsible for the quality of the candidate pool and the relevance of the job descriptions.

* Key Responsibilities: Uploading resumes, defining the "must-have" skills in Job Descriptions, triggering the Run Screening command, and validating the AI’s "Match Quality" scores.
* Access Level: Full access to Candidate and Job modules; limited access to system-wide billing or API settings.
* Dashboard Focus: "Match Quality," "Time Saved," and the "Efficiency Report."

### 3. Hiring Manager

The Hiring Manager acts as the final quality control. They do not usually engage in the "grunt work" of uploading files.

* Key Responsibilities: Reviewing the AI-generated "Top Matches" shortlist, providing feedback on candidate suitability to refine future screenings.

**1.User Registration & Login Module**

This is the gateway to the platform, ensuring that only authorized HR personnel and administrators can access sensitive candidate data.

* Functionality: Handles account creation, email verification, and secure authentication using JWT (JSON Web Tokens).
* Security: Implements Role-Based Access Control (RBAC). When a user logs in, the system checks their role (Admin, Recruiter, or Manager) to determine which dashboard features are visible.
* Integration: Supports Single Sign-On (SSO) for enterprise clients, allowing recruiters to log in using corporate credentials (e.g., Google Workspace or Microsoft Azure AD).

1. **SOS Trigger Module (System Override & Safety)**

In an AI-driven environment, an SOS Trigger acts as a fail-safe mechanism. This module monitors the "health" of the AI agent and the financial consumption of the project.

Functionality: Automatically pauses all active screenings if it detects an anomaly, such as:

* + Token Spike: If a single resume consumes more than a predefined limit of tokens (indicating a loop or parsing error).
  + Low Match Quality: If the AI scores fall below a $20\%$ threshold across an entire batch (indicating a mismatch between the JD and the candidate pool).

Manual Override: Allows Administrators to "Kill-Switch" the API calls instantly to prevent budget overruns.

**3. Helper Module (The AI Copilot)**

The Helper module serves as the "onboarding" and "optimization" assistant for the recruiter. It is designed to maximize the effectiveness of the screening agent.

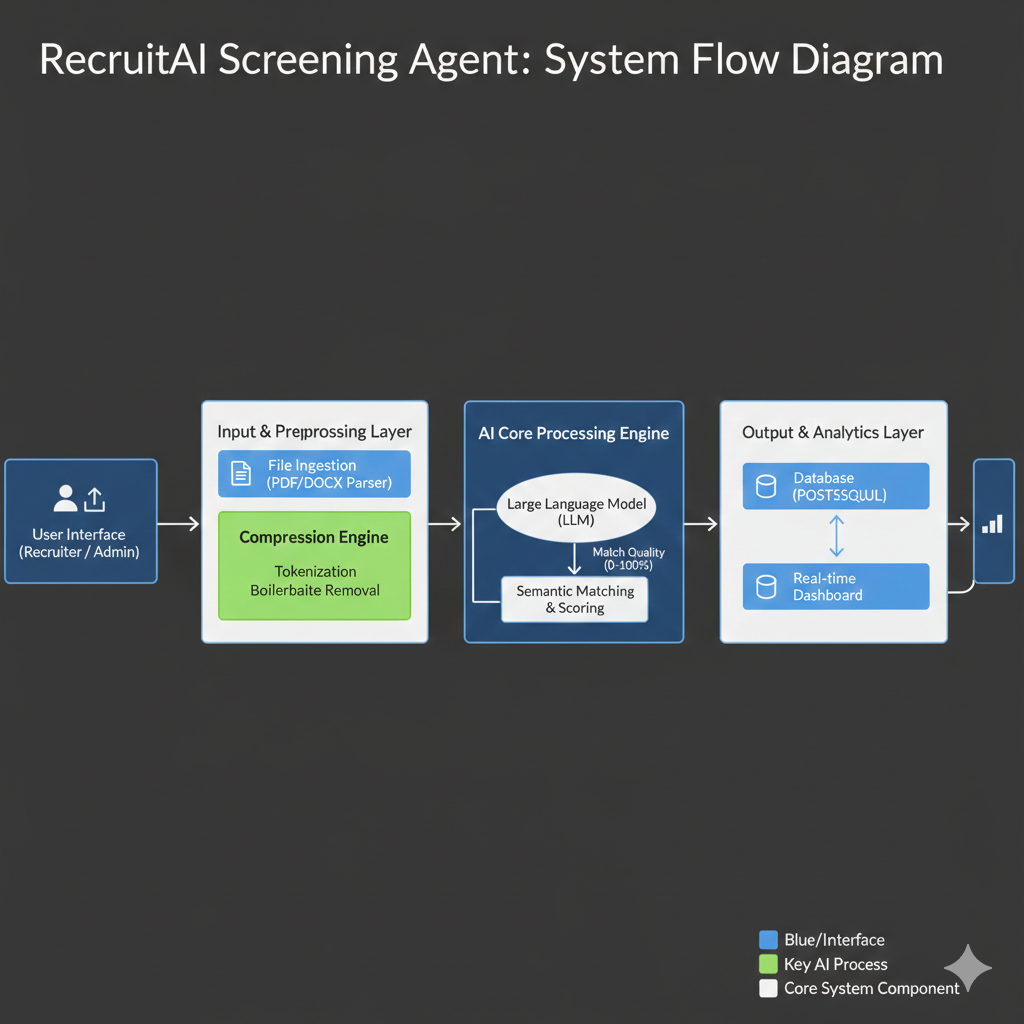
Functionality: \* JD Optimizer: Helps recruiters rewrite job descriptions to be "AI-readable," ensuring the semantic engine picks up the right signals.

* + Troubleshooter: Provides real-time tooltips and "Guided Tours" for new users navigating the dashboard for the first time.
  + Compression Preview: Shows the user a sample of what a "Compressed" resume looks like compared to the original, validating the logic of the engine.

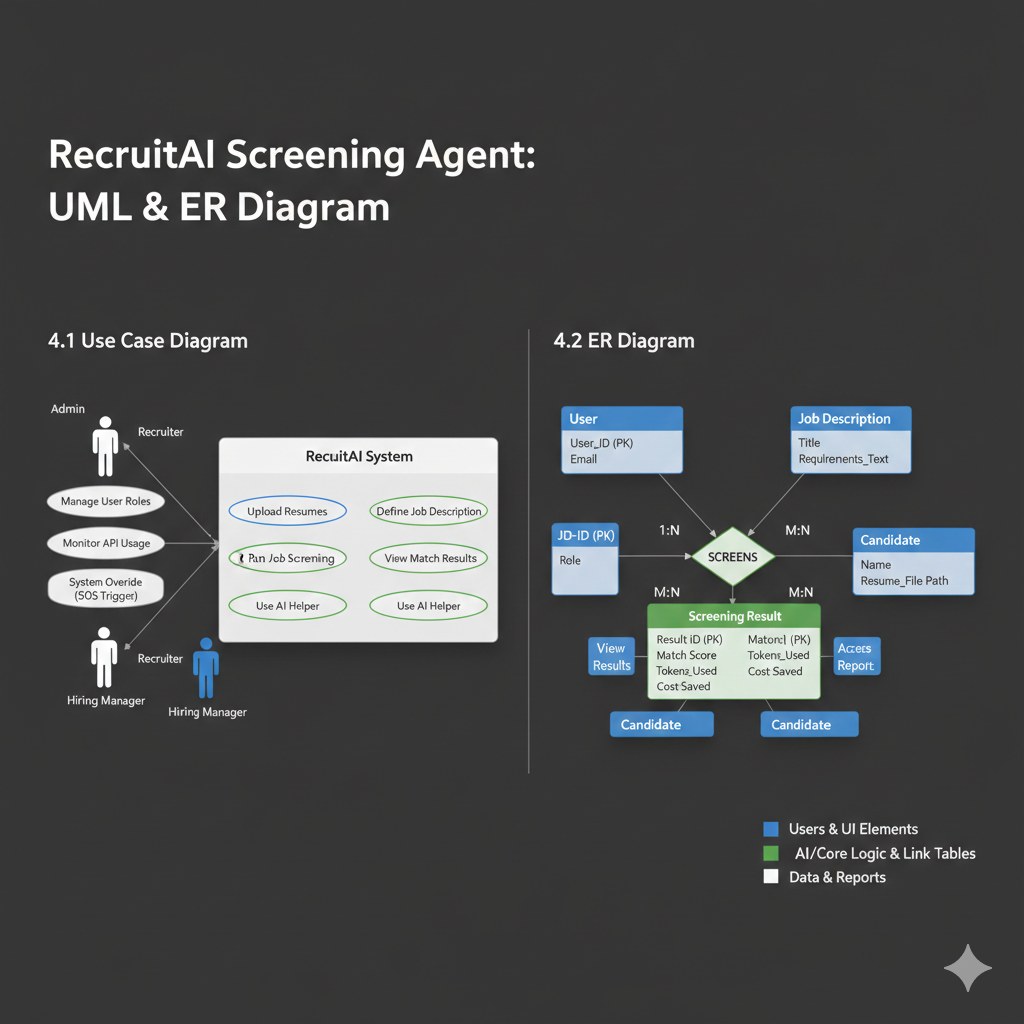
**Timeline Chart**



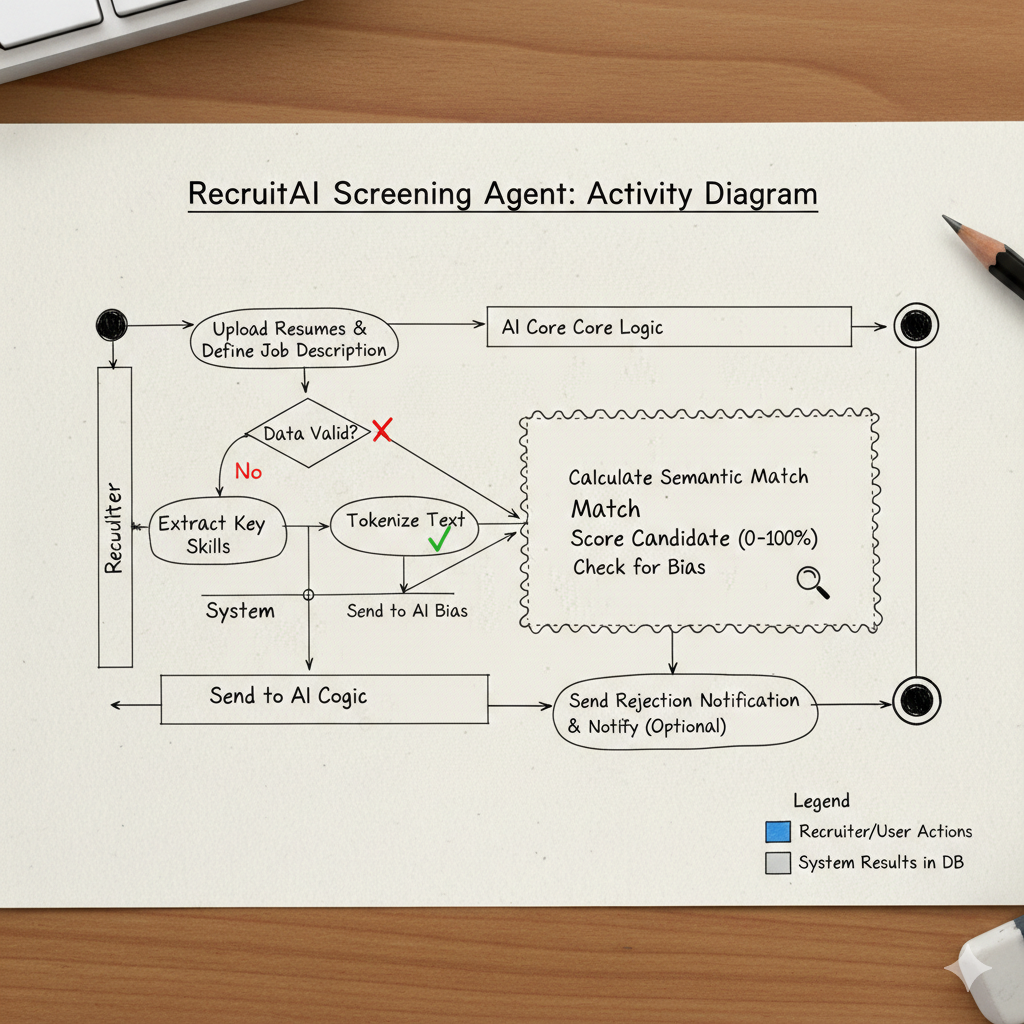
**System Flow Diagram**



**ER Diagram**



**Activity Diagram**



**References**

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* (The foundational dissertation for RESTful API design used in the system's backend).

Vaswani, A., et al. (2017). *"Attention Is All You Need."* (The seminal paper introducing the Transformer architecture which powers the LLMs used in the Screening Agent).

OpenAI (2024). *GPT-4 Technical Report.* (Documentation on tokenization and context window limits relevant to the Compression Engine).

Google Cloud (2025). *Gemini 1.5 Pro: Technical Documentation.* (Reference for semantic understanding and multi-modal processing capabilities).

International Organization for Standardization (2022). *ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE).* (Used to define Non-Functional Requirements).

Sommerville, I. (2016). *Software Engineering (10th Edition).* (Standard reference for creating Software Requirement Specifications (SRS) and UML modeling).

PostgreSQL Global Development Group (2025). *PostgreSQL 16.0 Documentation.* (Technical reference for relational database schema and ER diagram implementation).

React Documentation (2025). *React v19: Thinking in React.* (Guidelines for building the responsive dashboard UI).