

ResuMatch – AI-Powered Resume & Career Coach

Date: Sept 27, 2025
Version: 1.0



ResuMatch

Table of Contents

| | |
|---|----|
| <i>ResuMatch – AI-Powered Resume & Career Coach</i> | 1 |
| | 1 |
| Part A: Project Scope and Requirements | 3 |
| Section 1: Problem Statement..... | 3 |
| 1.1.a Problem & Need | 3 |
| 1.1.b List of Capabilities and Benefits..... | 3 |
| 1.2 Identify stakeholders and their roles..... | 4 |
| 1.3 Identify sub-systems | 4 |
| 1.4 Intended users of the SRS documentation | 4 |
| Section 2: General Overview Modelling..... | 5 |
| 2.1 Context Flow Diagram (CFD)..... | 5 |
| Section 3: Requirements – functional and non-functional..... | 5 |
| 3.1 Non-functional requirements | 6 |
| 3.2 Functional requirements..... | 6 |
| Section 4: Domain Class Diagram | 17 |
| 4.1 List of Classes | 17 |
| 4.2 Diagram | 17 |
| Section 5: ERD..... | 18 |
| Section 6: System Sequence Diagrams | 19 |
| 6.1 SSD – Optimize Resume | 19 |
| 6.2 SSD – Subscription | 19 |
| Section 7: State Machine Diagrams | 21 |
| 7.1 Subscription Object | 21 |
| 7.2 Resume Analysis Object | 21 |
| Section 8: Technologies..... | 23 |
| Section 9: Project Management (Gantt Chart) | 24 |

Part A: Project Scope and Requirements

Section 1: Problem Statement

1.1.a Problem & Need

Today, job seekers encounter significant difficulties when preparing resumes for job applications. Many are filtered out by Applicant Tracking Systems (ATS) due to missing keywords, unclear phrasing, and a lack of tailored alignment with job postings. This repetitive process leaves applicants with limited feedback, consumes valuable time, and often lowers confidence in their applications.

ResuMatch addresses these challenges by aiming to increase the number of interviews, produce clearer and more effective resumes, reduce tailoring time, measure progress, and ensure privacy and security throughout the process.

1.1.b List of Capabilities and Benefits

Capabilities

1. Upload Resume (PDF/DOCX).
2. Upload Job Posting (paste/URL/file).
3. AI Compare & Score (resume ↔ posting).
4. Keyword Gap Suggestions.
5. Rewrites & Tone Improvements.
6. Skill-Gap Report + training links.
7. Export Optimized Resume (DOCX/PDF).
8. Career Dashboard (history, trends).
9. Notifications (analysis ready, matches, billing).
10. Subscription/Billing.

Benefits

1. More Interviews.
2. Higher Resume Quality.
3. Career Guidance.
4. Skill Development.
5. Time Saved.
6. Confidence.

1.2 Identify stakeholders and their roles

- **Job Seeker (primary):** Non-technical writers; needs ATS-safe resumes.
- **Recruiter (tertiary):** Posts roles; receives sorted candidate suggestions.
- **Admin (internal):** Catalog, billing, reporting.
- **Career Coach (secondary/optional):** Manual review add-on.
- **External APIs:** Gemini (AI), Stripe (payments), SendGrid (emails).

1.3 Identify sub-systems

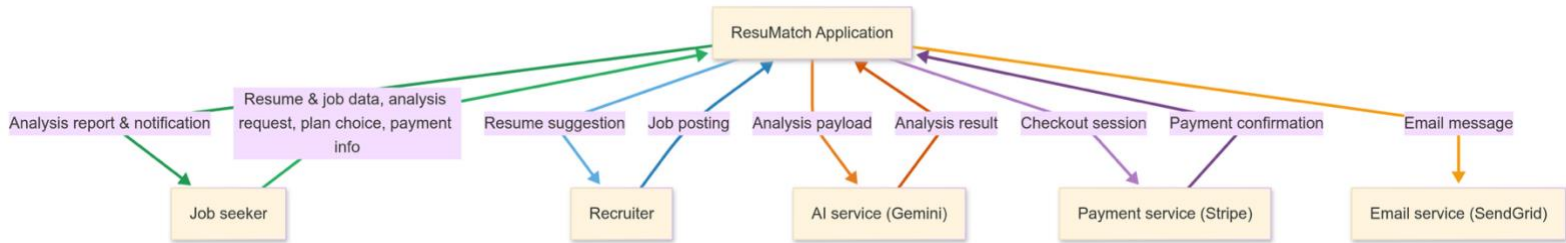
- Login & Registration Subsystem
- Resume Analysis Subsystem
- Subscription Subsystem
- Job Posting Subsystem
- Notification Subsystem

1.4 Intended users of the SRS documentation

- Developers/Software Design Architects
- Managers/Project Managers
- Software testers/Quality Assurance testers

Section 2: General Overview Modelling

2.1 Context Flow Diagram (CFD)



Section 3: Requirements – functional and non-functional

3.1 Non-functional requirements

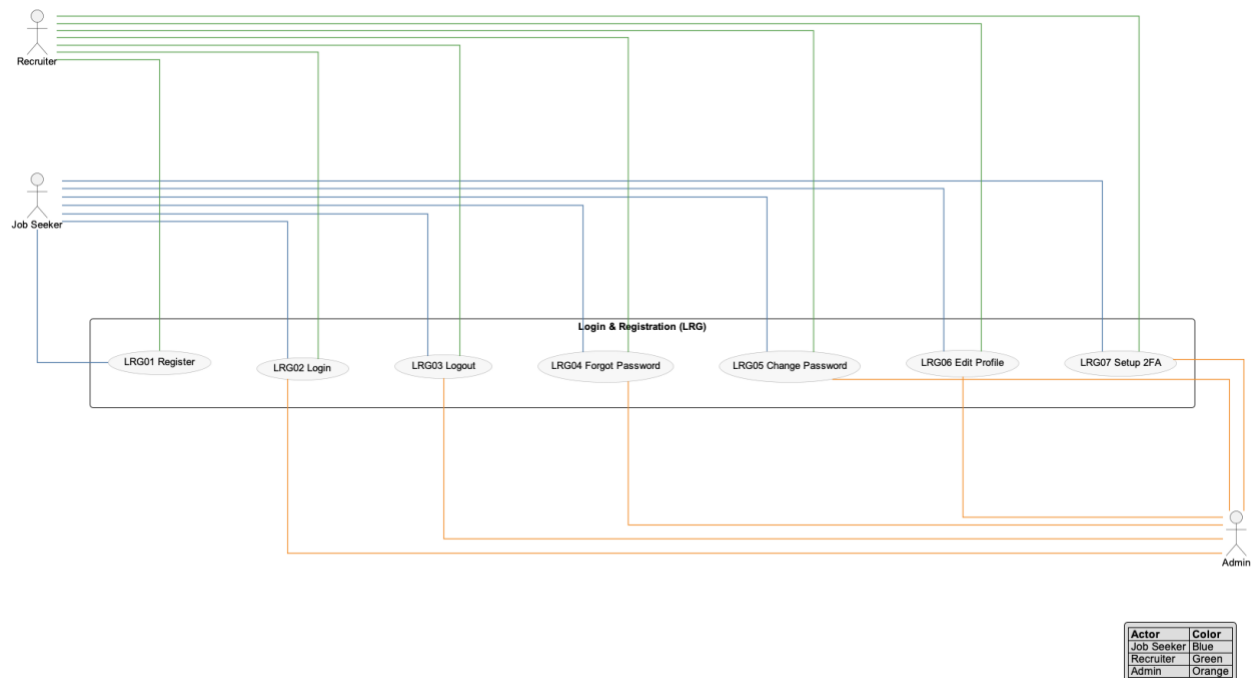
| NFR# | Name | Description |
|--------|----------------------------|--|
| NFR001 | Performance | AI analysis response $\leq 5s$ average for resumes $\leq 5MB$; export generation $\leq 3s$. |
| NFR002 | Security & Privacy | TLS 1.3 with HSTS; passwords hashed with PBKDF2/BCrypt/Argon2; MFA optional; data at rest encrypted with AES-256; GDPR rights supported; PCI-DSS scope isolated to Stripe. |
| NFR003 | Availability & Reliability | Uptime $\geq 99.5\%$ monthly; daily backups; RPO $< 24h$; RTO $< 8h$. |
| NFR004 | Usability & Accessibility | WCAG 2.1 AA compliance; keyboard-navigable; color-contrast checked; first-time user task (upload + analyze) ≤ 2 minutes median. |
| NFR005 | Maintainability | Unit test coverage $\geq 70\%$ on backend; API versioned; OpenAPI spec published. |
| NFR006 | Scalability | System can handle 100 concurrent analyses with queueing. |

3.2 Functional Requirements

3.2.1 Login & Registration Subsystem

| FR# | Name (Goal Use Case) | Role Player(s) | Description |
|----------|----------------------|------------------------------|---|
| FR LRG01 | Register | Job Seeker, Recruiter | Create an account with email/OTP verification. |
| FR LRG02 | Login | Job Seeker, Recruiter, Admin | Authenticate with credentials and optional 2FA. |
| FR LRG03 | Logout | All Users | End a session and invalidate tokens. |
| FR LRG04 | Forgot Password | All Users | Start reset flow with OTP/email link. |
| FR LRG05 | Change Password | All Users | Update password with enforced security rules. |
| FR LRG06 | Edit Profile | All Users | Update personal details and preferences. |
| FR LRG07 | Setup 2FA | All Users | Enable or disable multi-factor authentication. |

Use Case Diagram



User Story

1. As a Job Seeker, I want to register with my email and a strong password, so that I can create a secure account.

Acceptance Criteria:

- Verification email must be sent.
 - Account remains inactive until confirmed within 15 minutes.
 - Duplicate emails should show error within 2 seconds.
2. As a user, I want to log in securely, so that I can access my account.

Acceptance Criteria:

- Must reject invalid credentials within 2 seconds.
 - Must enforce lockout after 5 failed attempts.
 - Must support optional 2FA.
3. As a user, I want to reset my password when I forget it, so that I can regain access.

Acceptance Criteria:

- Reset token must be single use.
 - Must be valid for 10 minutes.
 - New password must differ from the last 5 passwords.
4. As a user, I want to edit my profile, so that my account reflects my latest information.

Acceptance Criteria:

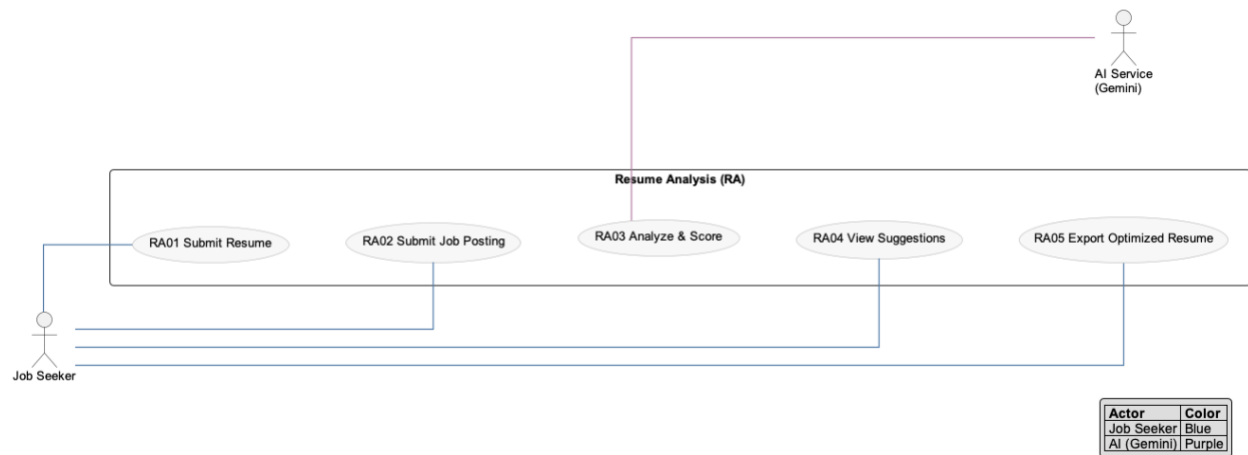
- Changes validated.
- Restrictions apply (e.g., name change once every 30 days).

3.2.2 Resume Analysis Subsystem

Functional Requirements Table

| FR# | Name (Goal Use Case) | Role Player(s) | Description |
|---------|-------------------------|----------------|---|
| FR RA01 | Submit Resume | Job Seeker | Upload resume (DOCX/PDF), parse content. |
| FR RA02 | Submit Job Posting | Job Seeker | Upload or paste job description. |
| FR RA03 | Analyze & Score | System ↔ AI | Compare resume vs job posting; compute match score. |
| FR RA04 | View Suggestions | Job Seeker | Display keyword gaps, rewrites, ATS warnings. |
| FR RA05 | Export Optimized Resume | Job Seeker | Generate/download updated DOCX/PDF. |

Use Case Diagram



User Story

1. As a Job Seeker, I want to upload my resume, so that the system can analyze it against job postings.

Acceptance Criteria:

- Accepts DOCX/PDF \leq 5MB.
- Confirms upload.
- Prevents duplicate resumes using hash.

2. As a Job Seeker, I want to see a score and suggestions, so that I can improve my resume.

Acceptance Criteria:

- Returns results in \leq 5 seconds.
- Provides score (0–100).
- Suggests missing keywords.
- Suggests rewrites, ATS flags.

3. As a Job Seeker, I want to export my optimized resume, so that I can use it in applications.

Acceptance Criteria:

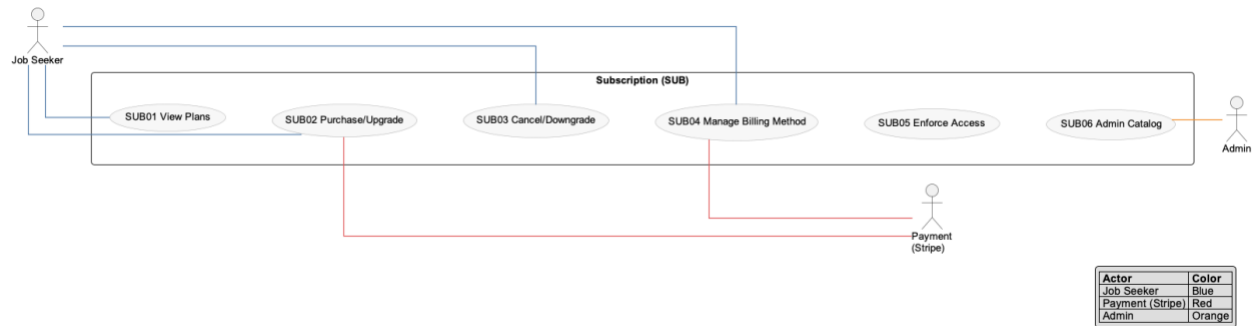
- File must include version tracking.
- Naming format: Firstname_Lastname_Role_vN.docx.

3.2.3 Subscription & Billing Subsystem

Functional Requirements Table

| FR# | Name (Goal Use Case) | Role Player(s) | Description |
|----------|-----------------------|---------------------|---|
| FR SUB01 | View Plans | Job Seeker | Display subscription tiers. |
| FR SUB02 | Purchase/Upgrade | Job Seeker ↔ Stripe | Subscribe securely via Stripe checkout. |
| FR SUB03 | Cancel/Downgrade | Job Seeker | Cancel or downgrade plans. |
| FR SUB04 | Manage Billing Method | Job Seeker ↔ Stripe | Update or remove payment methods. |
| FR SUB05 | Enforce Access | System | Enforce access to features based on plan. |
| FR SUB06 | Admin Catalog | Admin | Manage available plans, promo codes, reporting. |

Use Case Diagram



User Story

1. As a Job Seeker, I want to view available plans, so that I can choose the best option.

Acceptance Criteria:

- Must show Free, Basic, Premium with feature comparison.
2. As a Job Seeker, I want to subscribe securely, so that my payment details are protected.

Acceptance Criteria:

- Stripe-hosted checkout only.
 - System stores no card details.
 - Payment confirmation within 2 seconds.
3. As a user, I want to be blocked when I exceed my plan usage, so that I understand when I need to upgrade.

Acceptance Criteria:

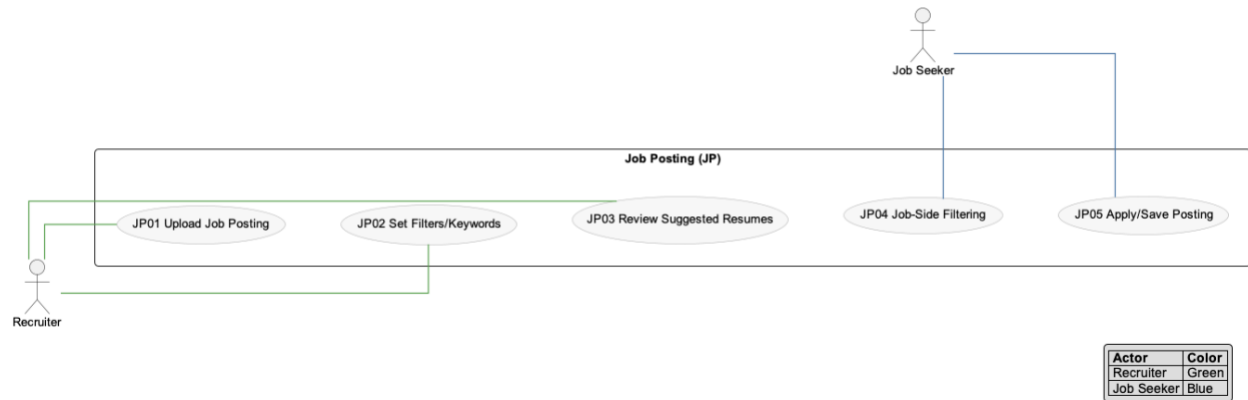
- Warnings at 80% of quota.
- Block at 100% with upgrade CTA.

3.2.4 Job Posting Subsystem

Functional Requirements Table

| FR# | Name (Goal Use Case) | Role Player(s) | Description |
|---------|--------------------------|----------------|--|
| FR JP01 | Upload Job Posting | Recruiter | Upload job descriptions in various formats. |
| FR JP02 | Set Filters/Keywords | Recruiter | Add requirements (skills, city, certifications). |
| FR JP03 | Review Suggested Resumes | Recruiter | Review AI-suggested resumes. |
| FR JP04 | Job-Side Filtering | Job Seeker | Filter postings by distance, salary, or skills. |
| FR JP05 | Apply/Save Posting | Job Seeker | Save postings or apply externally. |

Use Case Diagram



User Stories

1. As a Recruiter, I want to upload job postings, so that candidates can match against them.

Acceptance Criteria:

- Accepts DOCX, PDF, URL, or pasted text.
 - Validates format.
2. As a Recruiter, I want to review suggested resumes, so that I can quickly identify top candidates.

Acceptance Criteria:

- Displays ranked candidate list with match score.
 - Recruiter can accept/reject.
3. As a Job Seeker, I want to filter postings, so that I only see relevant opportunities.

Acceptance Criteria:

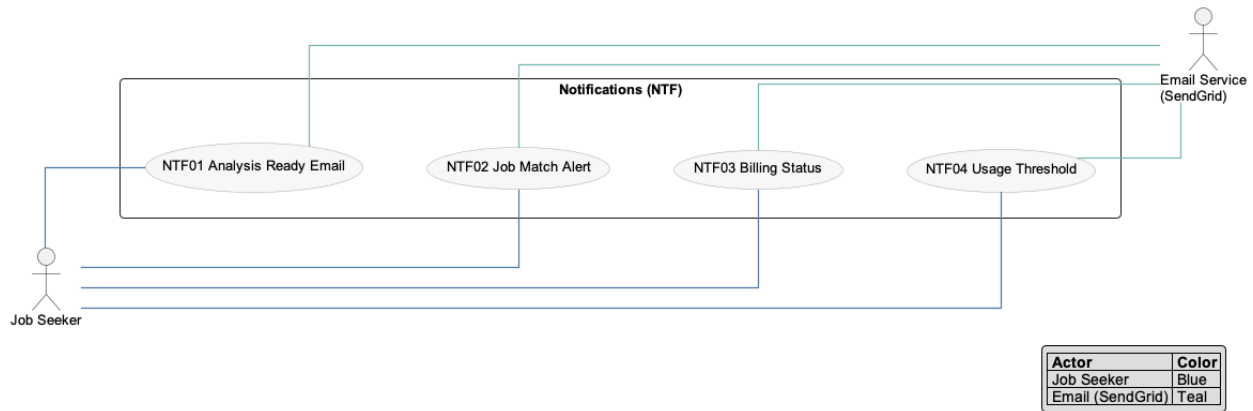
- Must allow filters (salary, distance, skills).
- Results must refresh in under 3 seconds.

3.2.5 Notification Subsystem

Functional Requirements Table

| FR# | Name (Goal Use Case) | Role Player(s) | Description |
|----------|----------------------|---------------------|--|
| FR NTF01 | Analysis Ready Email | System → Job Seeker | Notify when resume analysis completes. |
| FR NTF02 | Job Match Alert | System → Job Seeker | Notify when a new posting matches profile. |
| FR NTF03 | Billing Status | System → Job Seeker | Notify of payment success/failure/renewal. |
| FR NTF04 | Usage Threshold | System → Job Seeker | Notify when approaching plan limits. |

Use Case Diagram



User Stories

1. As a Job Seeker, I want to be notified when my resume analysis is ready, so that I don't have to keep checking manually.

Acceptance Criteria:

- Emails must send within 60 seconds of completion.
 - Retries 3 times if failed.
2. As a Job Seeker, I want to get billing and usage notifications, so that I'm always aware of my account status.

Acceptance Criteria:

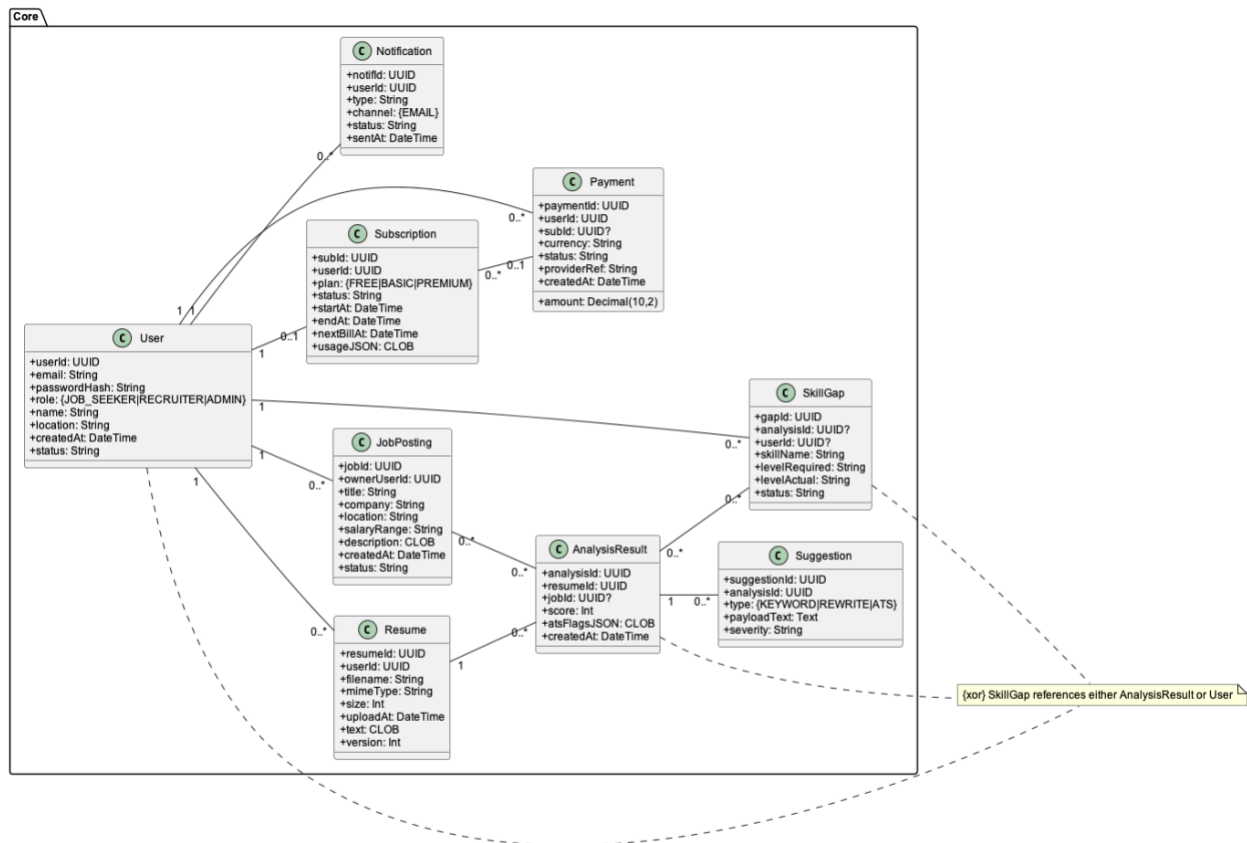
- Notifications must include clear subject, concise body, and direct link to relevant page.

Section 4: Domain Class Diagram

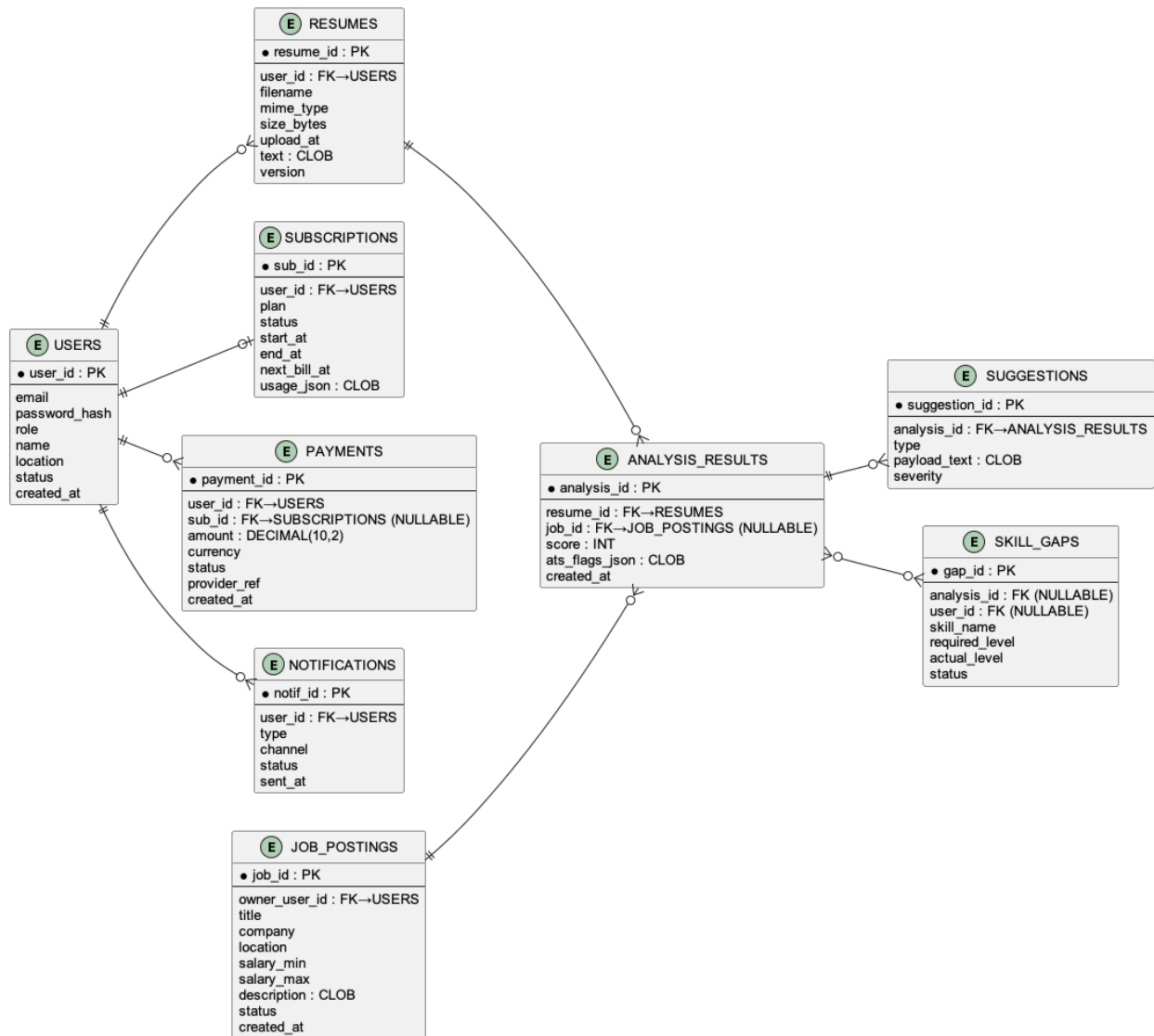
4.1 List of Classes

- User (userId, email, passwordHash, role, name, location, createdAt, status)
- Resume (resumeId, userId, filename, mimeType, size, uploadAt, text, version)
- JobPosting (jobId, ownerUserId, title, company, location, salaryRange, description, createdAt, status)
- AnalysisResult (analysisId, resumeId, jobId, score, atsFlagsJSON, createdAt)
- Suggestion (suggestionId, analysisId, type, payloadText, severity)
- SkillGap (gapId, analysisId or userId, skillName, levelRequired, levelActual, status)
- Subscription (subId, userId, plan, status, startAt, endAt, nextBillAt, usageJSON)
- Payment (paymentId, userId, subId, amount, currency, status, providerRef, createdAt)
- Notification (notifId, userId, type, channel, status, sentAt)

4.2 Diagram

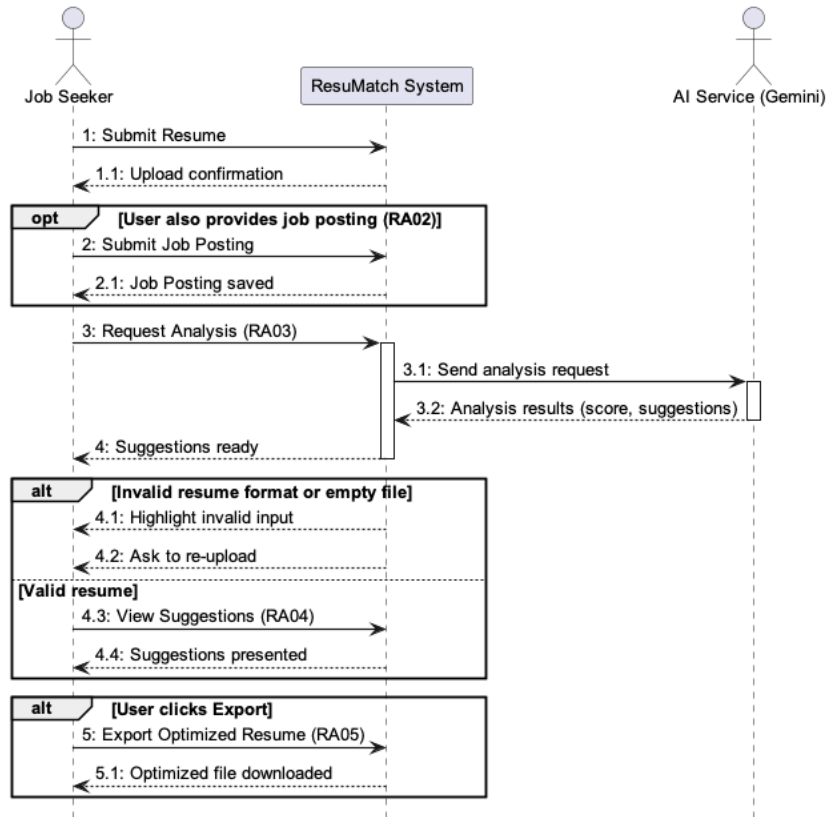


Section 5: ERD

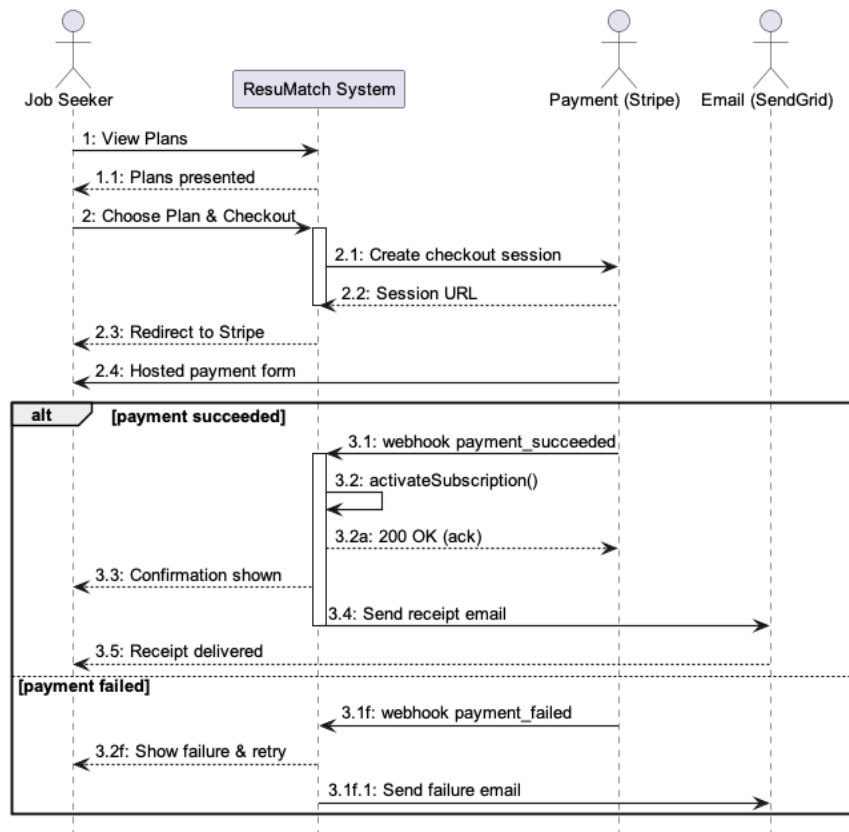


Section 6: System Sequence Diagrams

6.1 SSD – Optimize Resume

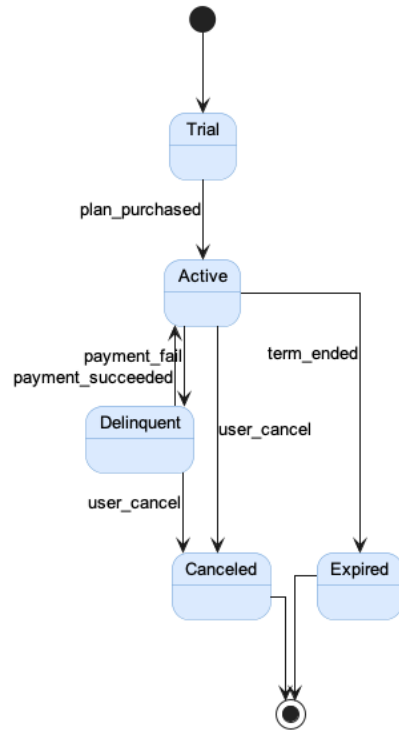


6.2 SSD – Subscription

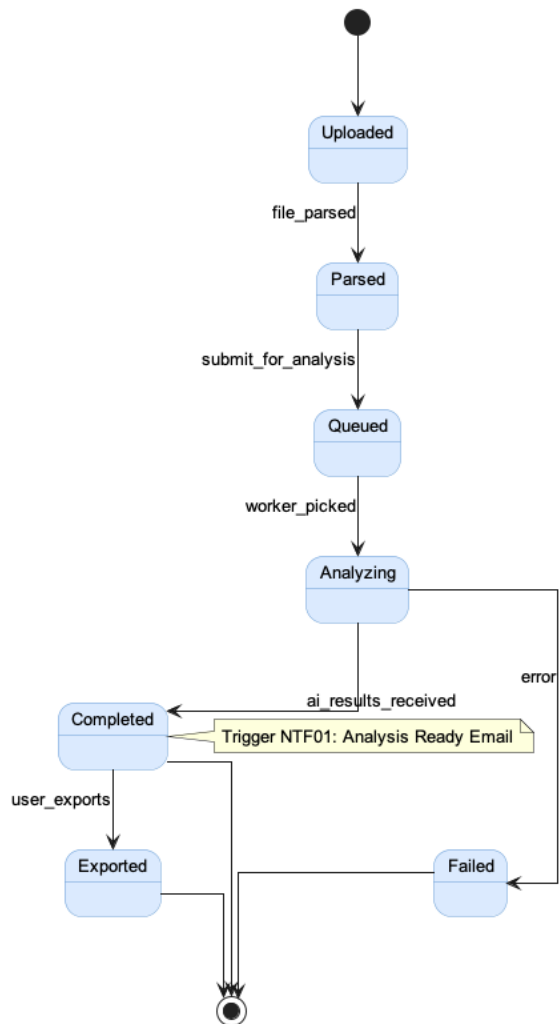


Section 7: State Machine Diagrams

7.1 Subscription Object



7.2 Resume Analysis Object



Section 8: Technologies

- **Frontend:** React (Vite/TypeScript), OAuth2/OpenID Connect.
- **Backend:** Python (FastAPI), Celery workers.
- **AI/NLP:** Google Gemini API.
- **Database:** Oracle SQL.
- **Payments:** Stripe.
- **Email:** SendGrid.
- **Hosting:** Cloud containers/VMs with HTTPS/TLS 1.3.

Section 9: Project Management (Gantt Chart)

