Software Requirements Specification

AI Job Source Agent

Version: 1.0
Date: 2025-02-13
Target Budget: $50/month
Build Tool: Claude Code

---

1. System Overview

Pipeline that extracts company career pages and job URLs from LinkedIn job listings.

Input: LinkedIn job search URL
Output: JSON file with company name, career page URL, position URL

---

2. Functional Requirements

2.1 LinkedIn Data Acquisition

- **FR-1.1**: Fetch job listings from LinkedIn via third-party API
- **FR-1.2**: Extract: company name, company website URL
- **FR-1.3**: Process 20-50 companies per execution
- **FR-1.4**: Handle API rate limits and errors gracefully

2.2 Career Page Discovery

- **FR-2.1**: Test common career page paths first (`/careers` , `/jobs` , etc.)
- **FR-2.2**: If direct paths fail, scrape homepage for career links
- **FR-2.3**: Search for keywords: "careers", "jobs", "join us", "opportunities"
- **FR-2.4**: Fallback to Claude API only when heuristics fail
- **FR-2.5**: Return absolute URLs only

2.3 Position Extraction

- **FR-3.1**: Navigate to career page (handle JavaScript rendering)
- **FR-3.2**: Extract first available job posting URL
- **FR-3.3**: Return absolute URL only
- **FR-3.4**: Timeout after 15 seconds per page

2.4 Data Output

- **FR-4.1**: Save results to JSON file with schema:

```
{
  "company_name": "string",
  "career_page_url": "string",
  "open_position_url": "string",
  "timestamp": "ISO8601"
}
```

- **FR-4.2**: Include execution statistics (success rate, API calls used)
- **FR-4.3**: Filename format: `job_sources_YYYY-MM-DD.json`

---

3. Non-Functional Requirements

3.1 Performance

- **NFR-1.1**: Process minimum 20 companies per run
- **NFR-1.2**: Complete execution within 10 minutes for 50 companies
- **NFR-1.3**: Max 5 seconds timeout per HTTP request

3.2 Cost Constraints

- **NFR-2.1**: LinkedIn API calls: <$25/month
- **NFR-2.2**: Claude API calls: <$20/month (max 50 calls/month)
- **NFR-2.3**: Prioritize free methods (>80% success via heuristics)

3.3 Reliability

- **NFR-3.1**: Log all errors with timestamps
- **NFR-3.2**: Continue execution on individual company failures
- **NFR-3.3**: Validate URLs before returning (200 status check)

3.4 Maintainability

- **NFR-4.1**: Modular design (5 separate modules)
- **NFR-4.2**: Environment variables for API keys
- **NFR-4.3**: Single configuration file for paths/patterns

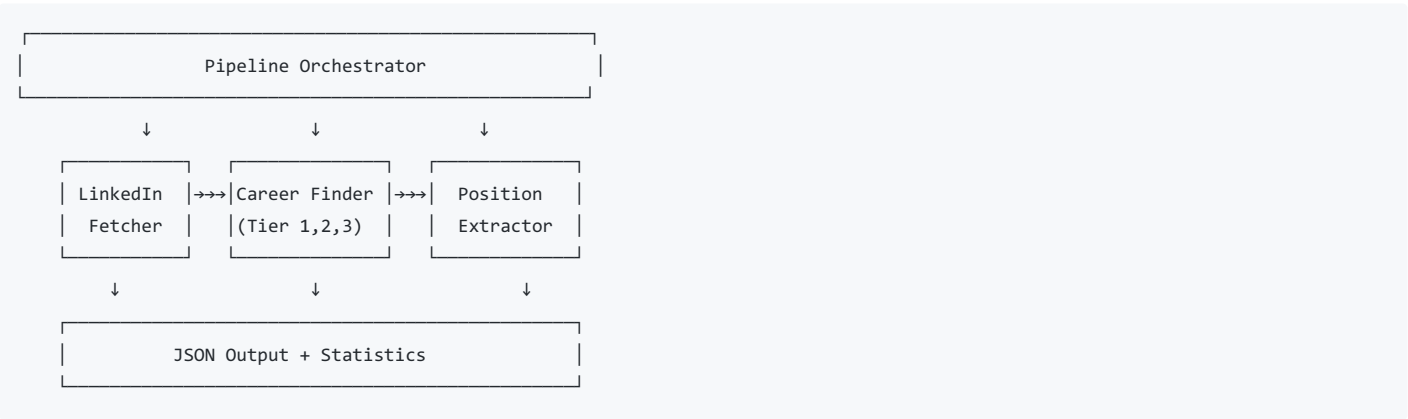
---

4. Technical Stack

Component	Technology	Rationale
LinkedIn API	Apify/RapidAPI	Pay-per-use, budget-friendly
Web Scraping	BeautifulSoup4	Lightweight HTML parsing
Browser Automation	Playwright	JS rendering, reliable
AI Fallback	Claude API (Sonnet 4.5)	Only when needed
Storage	JSON	Simple, portable
Language	Python 3.10+	Standard for automation

---

5. System Architecture



Career Finder Tiers:

1. **Tier 1** (FREE): Direct path testing (80% coverage)
2. **Tier 2** (FREE): Homepage scraping (15% coverage)
3. **Tier 3** ($$$): Claude API (5% coverage)

---

6. Module Specifications

6.1 Module: `linkedin_fetcher.py`

Input: LinkedIn job search URL, item limit
Output: List of dicts with company_name, company_url
Dependencies: `apify-client` or `requests`

API Budget: $20-25/month

6.2 Module: `career_finder.py`

Input: Company website URL
Output: Career page URL or None
Methods:

- `find_via_direct_paths()`
- `scrape_homepage()`
- `check_footer_nav()`
  Dependencies: `requests`, `beautifulsoup4`
  API Budget: $0

6.3 Module: `position_extractor.py`

Input: Career page URL
Output: First job posting URL or None
Dependencies: `playwright`
API Budget: $0

6.4 Module: `claude_fallback.py`

Input: Company website URL
Output: Career page URL or None
Rate Limit: 50 calls/month
Dependencies: `anthropic`
API Budget: $15-20/month

6.5 Module: `pipeline.py`

Input: LinkedIn URL, max_companies
Output: JSON file + console stats
Dependencies: All above modules + `json`

---

7. Configuration File

File: `config.py`

```
 # API Configuration
APIFY_TOKEN = env('APIFY_TOKEN')
ANTHROPIC_API_KEY = env('ANTHROPIC_API_KEY')

# Career Page Patterns
CAREER_PATHS = ['/careers', '/jobs', '/about/careers', ...]
CAREER_KEYWORDS = ['careers', 'jobs', 'opportunities', ...]

# Limits
MAX_CLAUDE_CALLS_PER_MONTH = 50
REQUEST_TIMEOUT = 5
BROWSER_TIMEOUT = 15000

# Output
OUTPUT_DIR = './output'
```

---

8. Error Handling

Error Type	Action
LinkedIn API failure	Log error, skip batch
Invalid company URL	Log, continue to next
Career page not found	Try Claude fallback → log if still fails

Error Type	Action
Position extraction timeout	Log, return partial data
Claude API limit reached	Log warning, skip Claude calls

9. Success Criteria

- ✅ Successfully process 50 companies in <10 minutes
- ✅ 85%+ success rate via free heuristics (Tier 1-2)
- ✅ <$50 monthly costs
- ✅ Valid JSON output with all required fields
- ✅ Modular codebase (each module <200 lines)

10. Out of Scope

- ❌ Real-time job monitoring
- ❌ Direct LinkedIn scraping (ToS violation)
- ❌ Job application automation
- ❌ Database persistence (JSON only)
- ❌ Web UI/dashboard
- ❌ Email notifications

11. Deliverables

1. 5 Python modules (`linkedin_fetcher.py` , `career_finder.py` , `position_extractor.py` , `claude_fallback.py` , `pipeline.py`)
2. `config.py` configuration file
3. `requirements.txt` with dependencies
4. `README.md` with setup instructions
5. Sample output: `job_sources_sample.json`

12. Environment Setup

```
# Required
Python 3.10+
pip install apify-client beautifulsoup4 playwright anthropic requests

# Environment Variables
export APIFY_TOKEN="your_token"
export ANTHROPIC_API_KEY="your_key"

# Playwright
playwright install chromium
```

13. Execution Command

```
python pipeline.py --linkedin-url "https://linkedin.com/jobs/search?keywords=engineer" --max 50
```

END OF SPECIFICATION