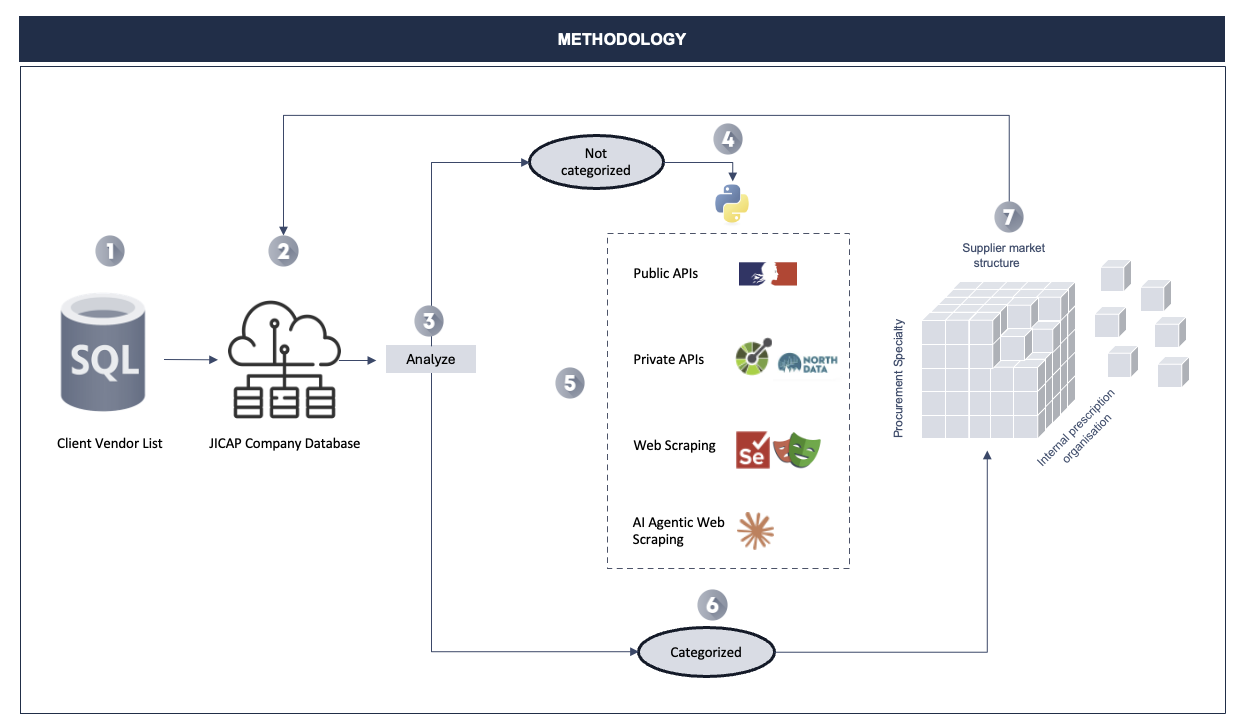
# **Product Requirements Document (PRD)**

## **JICAP Vendor Classification Automation System**

### **1. Product Overview**

**Product Name:** JICAP Vendor Classification System

**Version:** 1.0



**Target Users:** Non-technical business users, procurement teams, vendor management specialists

**Problem Statement:**

Organizations need to classify and categorize vendor information from client vendor lists by cross-referencing with existing databases and automatically fetching missing company data from various government APIs and web sources. The current manual process is time-consuming, error-prone, and requires technical expertise.

**Solution:**

A web-based application that automates vendor classification by processing uploaded vendor lists, cross-referencing with the JICAP Company Database, and automatically fetching missing company information from government APIs and web scraping sources.

### **2. Product Goals & Success Metrics**

**Primary Goals:**

* Reduce manual vendor classification time by 90%
* Eliminate human error in data entry and cross-referencing
* Provide a user-friendly interface for non-technical users
* Maintain data accuracy and consistency across vendor databases

**Success Metrics:**

* Processing time: < 5 minutes for 1000 vendor records
* User satisfaction score: > 4.5/5
* Data accuracy rate: > 98%
* System uptime: > 99.5%

### **3. User Stories & Requirements**

#### **3.1 Core User Stories**

**As a procurement manager, I want to:**

* Upload a client vendor list and have it automatically processed
* See which vendors are already classified vs. new
* Review all changes made to the database in a clear log
* Complete the entire process without technical knowledge

**As a data analyst, I want to:**

* Ensure data consistency across vendor databases
* Track processing history and changes
* Export updated databases for further analysis

#### **3.2 Functional Requirements**

**FR-1: File Upload & Processing**

* Support multiple file formats (.xlsx, .xlsb, .csv)
* File size limit: 50MB
* Automatic file validation and error handling
* Column mapping interface for non-standard file structures

**FR-2: Interactive Column Selection**

* Visual column preview with sample data
* Dropdown selection for vendor country and SIREN columns
* Data validation and format checking
* Clear error messages for invalid selections

**FR-3: Database Cross-Reference**

* Compare unique SIREN numbers against JICAP Company Database
* Identify new vs. existing vendors
* Process only unique SIREN numbers to avoid duplicates

**FR-4: Multi-Country Data Fetching**

* **France:** API integration with <https://recherche-entreprises.api.gouv.fr/search?q={siren>}
* **Belgium:** Web scraping from <https://kbopub.economie.fgov.be/kbopub/zoeknummerform.html>
* **Denmark:** Web scraping from <https://datacvr.virk.dk/enhed/virksomhed/{company_id>}
* Graceful handling of API failures and missing data

**FR-5: Database Updates**

* Update JICAP Company Database with new vendor information
* Preserve existing L1, L2, L3 classification columns
* Maintain data integrity and consistency

**FR-6: Progress Tracking & Logging**

* Real-time progress indicator during processing
* Detailed log file (log.txt) with all changes made
* Summary statistics (new vendors added, errors encountered)

#### **3.3 Non-Functional Requirements**

**NFR-1: Usability**

* Zero-code/low-code interface
* Maximum 3 clicks to complete core workflow
* Intuitive design following modern UX principles
* Mobile-responsive design

**NFR-2: Performance**

* Process 1000 records in < 5 minutes
* Concurrent processing for multiple data sources
* Efficient memory usage for large files

**NFR-3: Reliability**

* Automatic retry logic for API failures
* Data backup before modifications
* Transaction rollback capability for failed operations

**NFR-4: Security**

* Secure file upload handling
* Data encryption at rest
* No persistent storage of sensitive vendor data
* Session-based processing

### **4. Technical Architecture**

#### **4.1 Recommended Technology Stack**

* **Frontend:** Streamlit (Python-based web framework)
* **Backend:** Python with pandas for data processing
* **Web Scraping:** Playwright (more reliable than Selenium)
* **API Handling:** requests library
* **File Processing:** openpyxl, xlsxwriter
* **Deployment:** Docker containerization

#### **4.2 System Components**

**1. File Upload Module**

* Multi-format file parser
* Column detection and mapping
* Data validation engine

**2. Processing Engine**

* SIREN deduplication logic
* Country-specific data fetching orchestrator
* Database update manager

**3. Data Sources Integration**

* French Government API client
* Belgian web scraper (belgian\_siren.py)
* Danish web scraper (dk\_siren.py)
* Extensible architecture for additional countries

**4. Database Management**

* JICAP Company Database handler
* Change tracking and logging
* Backup and recovery system

**5. User Interface**

* Drag-and-drop file upload
* Real-time progress tracking
* Results visualization and export

### **5. User Experience Design**

#### **5.1 Workflow Overview**

1. **Upload:** Drag and drop Client Vendor List file
2. **Configure:** Select vendor country and SIREN columns
3. **Process:** Automatic processing with real-time progress
4. **Review:** Summary of changes and log file download
5. **Export:** Download updated JICAP Company Database

#### **5.2 Key UI Elements**

* **Dashboard:** Clean, minimal interface with clear CTAs
* **File Upload Zone:** Large, prominent drop area with format indicators
* **Column Mapper:** Side-by-side preview with dropdown selectors
* **Progress Tracker:** Visual progress bar with status updates
* **Results Panel:** Summary statistics and download links

### **6. Data Schema**

#### **6.1 Input Files**

**Client Vendor List (Variable columns):**

* Vendor Country (user-selected column)
* Vendor SIREN (user-selected column)
* Additional columns (ignored during processing)

**JICAP Company Database (8 columns):**

* Vendor Country
* Vendor SIREN
* Vendor Company Name
* Vendor Activity Code
* Vendor Activity Code Description
* L1 Classification (preserved)
* L2 Classification (preserved)
* L3 Classification (preserved)

#### **6.2 API Response Formats**

**French Government API:**

{  
 "results": [{  
 "nom\_complet": "Company Name",  
 "activite\_principale": "Activity Code",  
 "libelle\_activite\_principale": "Activity Description"  
 }]  
}

### **7. Error Handling & Edge Cases**

#### **7.1 File Processing Errors**

* Unsupported file formats → Clear error message with supported formats
* Corrupted files → File validation with repair suggestions
* Missing required columns → Column mapping guidance

#### **7.2 Data Processing Errors**

* Invalid SIREN numbers → Skip with logging
* API timeouts → Retry logic with exponential backoff
* Web scraping failures → Fallback mechanisms and manual review flags

#### **7.3 System Errors**

* Database connection issues → Automatic reconnection with user notification
* Memory limitations → Batch processing for large files
* Network connectivity → Offline mode with queued processing

### **8. Security & Compliance**

#### **8.1 Data Privacy**

* No persistent storage of uploaded files
* Session-based processing with automatic cleanup
* GDPR compliance for European vendor data

#### **8.2 Access Control**

* Role-based access (if multi-user deployment needed)
* Audit logging for all database modifications
* Secure file handling practices

### **9. Testing Strategy**

#### **9.1 Testing Scenarios**

* File format compatibility testing
* API integration testing with mock responses
* Web scraping reliability testing
* Database update integrity testing
* User interface usability testing

#### **9.2 Performance Testing**

* Large file processing (10K+ records)
* Concurrent user scenarios
* API rate limiting handling

### **10. Deployment & Maintenance**

#### **10.1 Deployment Options**

* **Local Deployment:** Docker container for on-premise use
* **Cloud Deployment:** Streamlit Cloud or similar PaaS
* **Enterprise Deployment:** Kubernetes cluster with load balancing

#### **10.2 Maintenance Requirements**

* Regular API endpoint monitoring
* Web scraping target validation
* Database backup scheduling
* Performance monitoring and optimization

### **11. Future Enhancements**

#### **11.1 Phase 2 Features**

* AI-powered activity classification (L1, L2, L3 automation)
* Additional country support (Germany, Netherlands, etc.)
* Batch processing scheduler
* Advanced reporting and analytics

#### **11.2 Integration Opportunities**

* ERP system integration
* Vendor management platform APIs
* Business intelligence dashboard connectivity

### **12. Risk Assessment**

#### **12.1 Technical Risks**

* **API Changes:** Government APIs may change structure → Regular monitoring and update procedures
* **Web Scraping Fragility:** Target websites may change → Automated testing and fallback mechanisms
* **Scale Limitations:** Large file processing → Implement batch processing and optimization

#### **12.1 Business Risks**

* **Data Accuracy:** Incorrect classification → Validation rules and manual review workflows
* **User Adoption:** Complex interface → Extensive user testing and feedback incorporation
* **Compliance:** Data privacy regulations → Legal review and compliance documentation

### **13. Success Criteria & Acceptance Tests**

#### **13.1 Acceptance Criteria**

* [ ] Upload and process 1000-record vendor list in < 5 minutes
* [ ] Correctly identify and classify 95%+ of valid SIREN numbers
* [ ] Generate accurate change logs for all database modifications
* [ ] Provide intuitive interface requiring no technical training
* [ ] Handle errors gracefully with clear user guidance

#### **13.2 Go-Live Requirements**

* Complete user acceptance testing
* Performance benchmarking completion
* Security audit approval
* User training materials and documentation
* Production deployment and monitoring setup

**Document Version:** 1.0

**Last Updated:** May 25, 2025