# Product Requirements - TCG eBay Batch Uploader

## 1. Introduction

This document outlines the requirements for an advanced application that automates listing Trading Card Game (TCG) inventory on eBay using AI-powered identification, intelligent multi-level caching, and comprehensive data enrichment. The application leverages state-of-the-art APIs for card identification and pricing, with direct eBay integration for professional listings.

**Primary Focus**: Pokémon TCG and Magic: The Gathering cards  
**Current Version**: 3.0 (Ultra-Cached, Async-Optimized, eBay EPS Integrated)

## 2. Goals

* **Minimize API costs** through intelligent multi-level caching (60-80% cost reduction achieved)
* **Maximize listing quality** through comprehensive data enrichment (all 40 eBay fields populated)
* **Optimize processing speed** through async processing (85% speed improvement achieved)
* **Ensure accurate pricing** by fetching only Near Mint condition prices
* **Enable efficient batch processing** of hundreds of cards with automatic image pairing
* **Provide quality control** through confidence scoring and review flags

## 3. Core Requirements

### 3.1. Data Ingestion & Processing

* **Input Format**: Process .jpg, .jpeg, .png, .webp, .bmp, and .tiff images from /Scans/ directory
* **Image Pairing**: Automatic detection of front/back sequential pairs (e.g., card001.jpg, card002.jpg)
* **Batch Processing**: Concurrent processing with configurable limits (default: 15 groups)
* **Processing Order**: Alphanumeric filename order for consistency
* **Memory Management**: Stream processing for large batches without memory overflow
* **Progress Tracking**: Real-time progress bars with ETA calculations

### 3.2. Multi-Level Intelligent Caching System

**Three-Tier Cache Architecture**:

1. **Image-Level Cache** (/output/ultra\_cache/images/):
   * Key: SHA256 hash of image content + file metadata
   * Value: Ximilar identification results
   * Purpose: Avoid re-processing identical image files
   * TTL: 30 days (configurable)
2. **Card-Type Cache** (/output/ultra\_cache/card\_data/):
   * Key: Normalized card name + set combination
   * Value: API pricing and metadata
   * Purpose: Reuse data for multiple copies of same card
   * TTL: 30 days (configurable)
3. **eBay EPS Cache** (/output/ultra\_cache/ebay\_eps/):
   * Key: Image hash
   * Value: eBay-hosted image URL
   * Purpose: Avoid re-uploading identical images
   * TTL: 30 days (matches eBay retention)

**Cache Features**:

* LRU eviction with configurable size limits (default: 10GB total)
* Persistent storage surviving application restarts
* Performance metrics and hit rate reporting
* Automatic cleanup of expired entries

### 3.3. AI-Powered Card Identification

**Ximilar API Integration**:

* Visual card identification with confidence scoring
* Extracts: name, number, set, rarity, game type
* Handles special characteristics (1st Edition, Shadowless, etc.)
* Confidence thresholds:
  + High: ≥95% (automatic processing)
  + Medium: 85-95% (process with review flag)
  + Low: <85% (flag for manual review)

### 3.4. Comprehensive Market Pricing

**Pokemon TCG API**:

* **Near Mint prices only** (market or mid price fields)
* Smart category selection based on card characteristics
* Handles special editions with price mappings
* Direct TCGPlayer URL extraction

**Scryfall API (MTG)**:

* **Near Mint prices only** (default Scryfall pricing)
* Foil vs non-foil detection
* Comprehensive MTG-specific data
* Direct TCGPlayer URL extraction

**Pricing Strategy**:

* 30% markup applied to all prices (configurable)
* Minimum price floor: $1.99
* Default price: $5.00 for cards without market data

### 3.5. eBay Integration

**eBay EPS (Picture Services)**:

* Direct image upload to eBay's CDN
* Multipart form handling with retry logic
* 30-day retention with automatic URL caching
* Support for multiple images per listing

**Business Policies**:

* Payment: Immediate Payment Required
* Shipping: Price-based selection (<$20 vs ≥$20)
* Returns: 30-day return window
* All policies configurable via environment variables

### 3.6. Data Enrichment & Field Population

**All 40 eBay Fields Populated**:

**Core Fields**:

* Action, Category (183454), Title, Description
* Condition (4000 - Near Mint), Format (FixedPrice)
* Duration (GTC), StartPrice, Quantity (1)
* Location, PostalCode, DispatchTimeMax

**Item Specifics**:

* Game (with "TCG" suffix)
* Card Name, Character (first word of name)
* Set, Rarity, Card Number
* Graded (No), Manufacturer
* Language, Finish, Features
* Card Size, Year Manufactured, Vintage status
* Country/Region of Manufacture
* Defense/Toughness (MTG only)
* HP, Card Type, Attribute/MTG:Color

**Enhanced Fields**:

* Professional HTML descriptions
* SEO-optimized titles (80 char limit)
* Direct eBay EPS image URLs
* Confidence scores and review flags

### 3.7. Title Generation

**Format**: [Game] {Card Name} {Number} {Set} {Rarity} {Finish} {Characteristics} NM/LP {Language}

**Rules**:

* Always start with game prefix (Pokémon/MTG)
* Skip redundant terms (Normal, Regular, English)
* Include meaningful terms only
* Truncate intelligently at 80 characters
* Add "JP" for Japanese cards only

### 3.8. Output Requirements

**Excel Format** (.xlsx):

* Filename: tcg\_listings\_YYYYMMDD\_HHMMSS.xlsx
* Location: /output/ folder
* Sheet: "eBay\_Upload" with all 40 columns
* Conditional formatting for review items
* Color-coded confidence scores

**Review System**:

* OK: Successfully processed
* LOW\_CONFIDENCE: Below threshold
* MISSING\_DATA: Incomplete information
* OCR\_ERROR: Potential misidentification

### 3.9. Performance Requirements

**Speed Targets**:

* 25-50 cards per minute
* 15 concurrent image groups
* 25 concurrent API calls max
* Sub-second cache lookups

**Reliability**:

* Exponential backoff retry
* Graceful API failure handling
* Resume capability for interruptions
* Comprehensive error logging

**Cost Optimization**:

* 60-80% API cost reduction via caching
* Batch processing optimization
* Real-time cost tracking
* Cache efficiency reporting

## 4. Configuration

### 4.1. Environment Variables (.env)

All sensitive configuration via environment variables:

* API keys (Ximilar, Pokemon TCG, eBay, OpenAI)
* Processing limits and thresholds
* Rate limiting parameters
* Business policy names
* Cache configuration

### 4.2. Fallback Configuration

config.json as fallback for non-sensitive settings:

* Default processing parameters
* URL endpoints
* File paths

## 5. Quality Assurance

### 5.1. Data Validation

* Card name verification
* Price reasonableness checks
* Image quality validation
* Completeness verification

### 5.2. Error Handling

* API failure recovery
* Network timeout handling
* Invalid data graceful degradation
* User-friendly error messages

### 5.3. Monitoring

* Processing success rates
* Cache hit ratios
* API usage tracking
* Performance metrics

## 6. Security & Compliance

* No storage of sensitive data in code
* Environment variable usage for credentials
* eBay API compliance
* Rate limiting respect
* Copyright compliance for descriptions

## 7. Future Enhancements

* Visual similarity detection
* Direct eBay API integration
* Condition assessment AI
* Multi-language support expansion
* Automated repricing
* Inventory tracking integration