404 Not Found

Digital Newspaper Website
Use-Case Specification: Check Plagiarism

Version 1.0

Revision History

Date	Version	Description	Author
18/07/2025	1.0	Complete the document	Lê Đình Minh Quân

Table of Contents

Use-Case Specification: Check Plagiarism

1. Use-Case Name

1.1 Brief Description

The *Check Plagiarism* use case describes how an authorized User (e.g., Writer, Editor, or Admin) submits an article draft (or selected text) to the plagiarism detection service. The system analyzes the content against internal repositories, previously published articles, and external sources (via integrated plagiarism API / search indexes) and returns a similarity report with matched segments, sources, and an overall similarity percentage. The outcome guides editorial decisions (revise, cite sources, reject, or approve).

2. Flow of Events

2.1 Basic Flow

Actor: Writer or Editor (authenticated; has permission to run plagiarism checks)
Preconditions: Draft article exists in the system and is accessible to the Actor; plagiarism service is operational; content length within allowed limits.
Trigger: Actor initiates a plagiarism check (e.g., clicks **"Check Plagiarism"** or saves draft with auto-check enabled).

1. Actor opens the draft and selects **Check Plagiarism**.
2. System validates prerequisites: user permission, draft state (not empty), size limits.
3. System normalizes content (strip markup, unify line endings, lowercase as needed, remove stop sections like boilerplate disclaimers per policy).
4. System packages normalized text with metadata (document ID, language, author ID, timestamp) and submits request to the plagiarism engine.
5. System displays a *"Scanning for similarity…"* progress indicator (with optional estimated time if > few seconds).
6. Plagiarism engine tokenizes and compares content against: (a) internal published corpus, (b) internal draft archive (other users' drafts if permissible), (c) cached external index results / third-party API.
7. Engine produces a similarity report (overall percentage, list of matched passages, source identifiers, match confidence, match types).
8. System stores the raw report and derives a summarized view (highlight spans, aggregated percentages by source).
9. System displays the plagiarism results to the Actor: overall similarity %, critical matches flagged above configured threshold(s), color-highlighted overlapping segments.

10. System logs an audit event `{userId, docId, action=PLAGIARISM_CHECK, similarity=<value>, timestamp}`.

11. Actor reviews results and decides next action (revise, cite, request exemption, or mark as clean). Use case ends on decision.

2.2 Alternative Flows

2.2.1 Large Document – Asynchronous Processing
Point of deviation: Step 4.

1. If content size exceeds synchronous threshold, system queues job and returns a "Scan queued" status.

2. Actor sees queued indicator and may continue editing.

3. Background worker completes analysis and updates report.

4. System notifies Actor (in‑app notification or email) → Actor opens report (resume at Basic Flow Step 9).

2.2.2 External Service Timeout / Unavailable
Point of deviation: Step 6.

1. External API times out or returns 5xx.

2. System retries per backoff policy (e.g., 2 more attempts).

3. If still failing, system returns partial internal corpus results + message "External sources temporarily unavailable."

4. Actor may retry manually later (return to Step 1) without losing draft state.

2.2.3 Insufficient Permissions
Point of deviation: Step 2.

1. User lacks required role/permission.

2. System denies request, shows "You are not authorized to perform plagiarism checks."

3. Audit event logged with outcome=DENIED.

4. Use case ends (no scan performed).

2.2.4 Content Too Short / Too Long

Point of deviation: Step 2 or 3.

1. Content length below minimum tokens (e.g., < 50 words) or above maximum (e.g., > 20,000 words).
2. System displays validation message specifying limit and prevents submission.
3. Actor adjusts content length and re-initiates (return to Step 1).

2.2.5 Duplicate Pending Scan
Point of deviation: Step 4.

1. An identical hash for this draft is already processing.
2. System merges request or informs Actor "A scan is already in progress."
3. Actor waits or views pending status (resume at Step 9 once complete).

2.2.6 High Similarity Threshold Breach (Critical Alert)
Point of deviation: Step 9.

1. Overall similarity or single-source match exceeds critical threshold (e.g., > 40%).
2. System flags report as **Critical** and (if configured) notifies Editor / Admin.
3. Actor must choose between revising or requesting formal review before publication (publication workflow blocked until resolved).

2.2.7 Exact Self-Match Only
Point of deviation: Step 7.

1. All matches are the author's previously published articles (self-plagiarism).
2. System categorizes these separately (self-matches) and may reduce their weighting per policy.
3. Actor views adjusted similarity (rejoins Step 9).

2.2.8 Significant Quoted / Properly Cited Material
Point of deviation: Step 8 / 9.

1. Large portions flagged but enclosed in quotes or have citations (e.g., references or block quotes).
2. System applies citation detection rules to discount properly cited spans from final critical percentage (but still shows them).

3. Revised similarity displayed to Actor (Step 9).

2.2.9 Language Detection Mismatch
Point of deviation: Step 3.

1. Detected language differs from expected or mixed language detected.
2. System switches or augments comparison indexes accordingly.
3. Flow resumes at Step 4 without user interruption.

2.2.10 Policy Update During Scan
Point of deviation: Step 6.

1. Internal similarity thresholds changed mid‑scan (e.g., Admin updates config).
2. System re‑applies new thresholds to results before presenting Step 9.
3. Actor sees badge “Threshold updated during scan” for transparency.

2.2.11 Manual Cancellation by Actor
Point of deviation: Step 5 (asynchronous or long scan).

1. Actor clicks **Cancel Scan**.
2. System attempts to abort queued job; if already running, marks result stale on completion.
3. System logs cancellation outcome and informs Actor.
4. Use case ends (no report delivered unless job already finished early).

2.2.12 Integrity Check Failure (Tampered Content)
Point of deviation: Step 4 or 6.

1. Content hash on completion does not match original submission hash (draft changed mid processing without snapshot).
2. System flags integrity error, discards report, and requests a new scan.
3. Actor re-initiates (Step 1) after confirming stable draft.

2.2.13 Risk Engine Anomaly (Unusual Source Pattern)
Point of deviation: Step 7 / 8.

1. Many short matches from diverse unrelated sources trigger risk heuristic.
2. System elevates severity, optionally prompts for manual editorial review.
3. Actor cannot mark as “Clean” until review resolution (then resumes Step 11 decision).

2.2.14 Backup Source Index Offline

Point of deviation: Step 6.

1. Secondary (supplemental) index offline; primary still available.
2. System returns report with note “Some secondary sources unavailable.”
3. Actor may accept partial report or schedule re-scan.

2.2.15 Automatic Re-Scan on Revision

Point of deviation: After Step 11 (post decision).

1. Actor modifies content above configured change threshold (e.g., >15% new tokens) within a freshness window.
2. System triggers auto re-scan (notify Actor) and updates report (returns to Step 9).

3. Special Requirements

- **Accuracy:** Matching must minimize false positives via token n-gram overlap + fuzzy matching (stemmed / case-insensitive); properly cited/quoted sections discountable.
- **Performance:** Synchronous scans ≤ 8 seconds 95th percentile for average article length (e.g., ≤1500 words); async path handles larger documents.
- **Scalability:** Horizontal scaling for parallel scans; queue throughput target (e.g., 50 scans/minute baseline).
- **Security:** Submitted text transmitted over TLS; stored reports encrypted at rest; external API keys secured.
- **Privacy:** Do not expose private drafts from other authors unless access policy permits; anonymize user IDs in external requests if required.
- **Auditability:** Log each scan with docId, userId, similarity percentage, source counts, outcome (SUCCESS / PARTIAL / FAILED / CANCELED).
- **Configurability:** Thresholds (warning %, critical %, self-match discount) adjustable without redeploy.
- **Internationalization:** Multi-language tokenization & stemming; UI messages localizable.

- **Accessibility:** Results highlighting navigable via keyboard; color distinctions paired with text labels.
- **Reliability:** Retry logic for transient external failures; gracefully degrade to internal-only results.
- **Integrity:** Snapshot hash ensures the report matches the analyzed content version.
- **Legal / Compliance:** Retain reports for defined retention period (e.g., 1 year) then purge per policy.

4. Preconditions

- User authenticated and authorized for plagiarism checks.
- Draft content saved (not empty) and within permissible size range.
- Plagiarism engine and required indices reachable.
- Configured thresholds and policies loaded successfully.

5. Postconditions

- **Success:** Similarity report stored; Actor informed; decision pending editorial action.
- **Partial:** External sources missing; internal results delivered with notice.
- **Failure:** Scan aborted (timeout, integrity error); no final report; Actor prompted to retry.
- **Canceled:** User aborted; any late result marked stale.
- **Critical:** High similarity flagged; publication workflow gated pending remediation.
- **Auto Re-Scan Scheduled:** Content change triggers queued new scan.

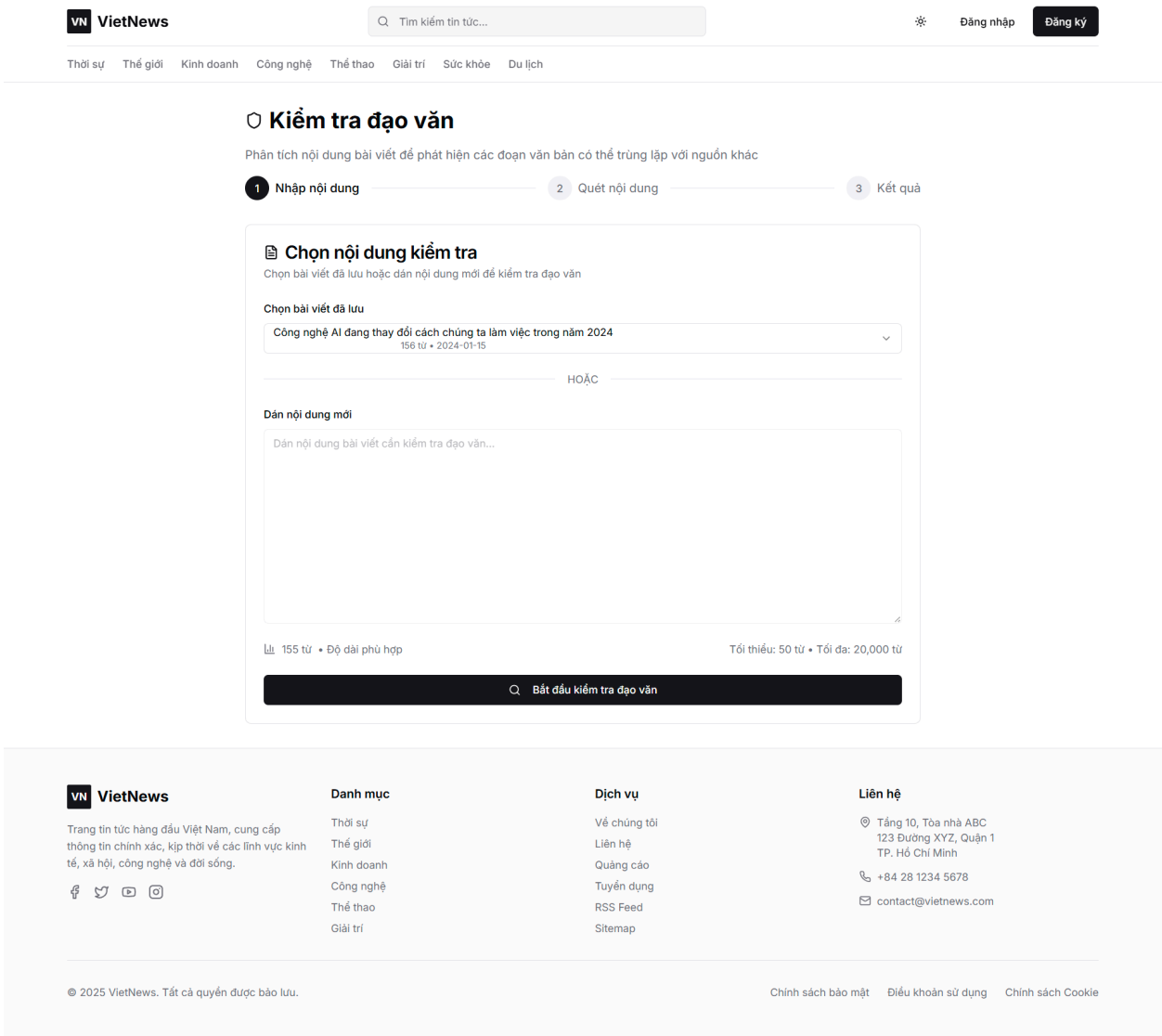
6. Extension Points

- **Auto Citation Suggestions: After Step 9, propose citation formatting for unmatched raw quotations.**
- **Machine Learning False Positive Filter: Between Steps 7–8 reclassify borderline matches.**
- **Similarity Trend Analytics: After Step 10, aggregate metrics for dashboards (average similarity per author).**
- **Inline Rewrite Assistant: Post Step 9 launches a guided paraphrasing assistant for flagged passages.**

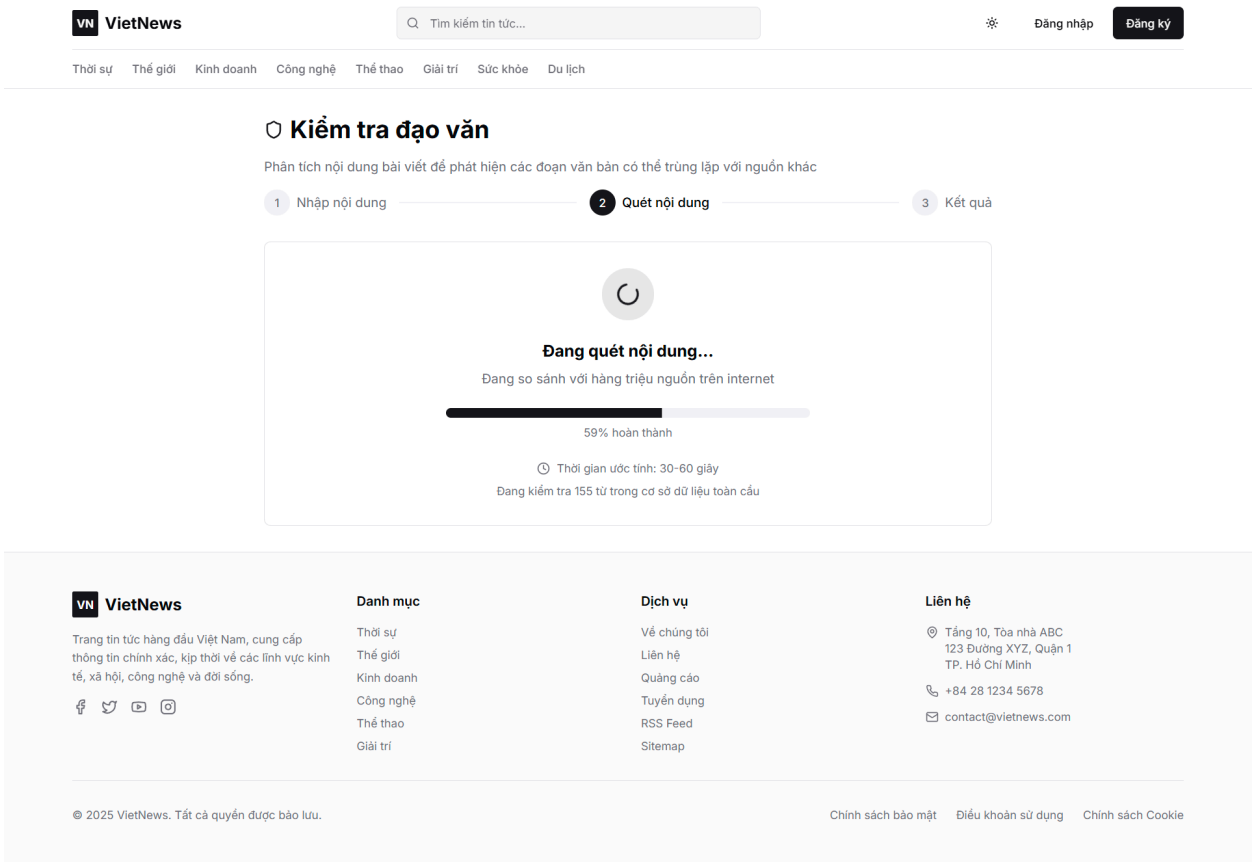
7. Low Fidelity Prototype For Use Case

Prompt for AI tools: Enhance the /test/plagiarism-report pages of my Digital Newspaper Website to simulate realistic, step-by-step processes. Use the same layout, design system, and UX style as the rest of the website. Include all typical stages described in the use-case specifications.📄 /test/plagiarism-report – Plagiarism Checking SimulationCreate a full plagiarism detection workflow:Step 1 – Draft Input:Allow selecting a saved draft or pasting article content.Validate length: warn if too short (<50 words) or too long (>20,000).Include “Run Check” button.Step 2 – Scanning:Show loading screen: “Scanning for similarity…” with animated progress.Simulate delay or mock API response.Step 3 – Results Display:Show:Overall similarity score (e.g., 32%)List of matching sources with confidence barsHighlighted matched segments in article contentUse different colors for critical (red), warning (orange), and clean (green) matches.Step 4 – Decision:If score is high, show alert and block submission.Buttons: “Revise”, “Request Review”, “Mark as Clean”.On acceptably low score, show: “Check passed. You may submit to Editor.”

404 Not Found



404 Not Found



404 Not Found