

KẾ HOẠCH TRIỂN KHAI DỰ ÁN Blog & Video Platform

TUẦN 0: PHÂN TÍCH & THIẾT KẾ Business Analysis Document

Phiên bản:	1.0
Ngày tạo:	27/11/2025
Tác giả:	Claude & Development Team
Loại tài liệu:	Technical Planning Document

MỤC LỤC

1.	PHẦN A: USER FLOW DIAGRAMS	4
	1.1 Guest → View Content Flow	4
	1.2 User Registration → First Post Flow	5
	1.3 Author → Create Post with Video Flow	6
	1.4 User → Search & Bookmark Flow	7
	1.5 Admin → Content Moderation Flow	8
2.	PHẦN B: USE CASES CHI TIẾT	9
	2.1 UC-01: Đăng ký Tài khoản	9
	2.2 UC-02: Tạo Bài viết Mới	11
	2.3 UC-03: Upload và Encode Video	14
	2.4 UC-04: Tương tác (Comments, Likes, Bookmarks)	18
	2.5 UC-05: Tìm kiếm với AI	22
	2.6 UC-06: Thống kê & Analytics	24
3.	PHẦN C: BUSINESS RULES VALIDATION	27
	3.1 UC × BR Matrix	27
	3.2 Chi tiết từng Business Rule	28
	3.3 Coverage Report	35
	3.4 Implementation Checklist	36
	3.5 Traceability Matrix	38

PHẦN A: USER FLOW DIAGRAMS

Phần này trình bày 5 luồng nghiệp vụ chính của hệ thống dưới dạng flowcharts. Mỗi flow diagram minh họa các bước người dùng thực hiện và cách hệ thống phản hồi, bao gồm cả các trường hợp lỗi và alternative flows.

1.1 Flow 1: Guest → View Content Flow

Mô tả: Luồng này mô tả hành trình của Guest (khách chưa đăng nhập) khi truy cập website. Guest có thể xem nội dung nhưng không thể tương tác (comment, like, bookmark). Khi muốn tương tác, hệ thống sẽ hiển thị popup yêu cầu đăng nhập.

Các bước chính:

- Guest vào trang web → Xem Homepage
- Chọn hành động: Xem bài viết, Xem video, hoặc Tìm kiếm
- Khi muốn comment → Hiện popup: "Đăng nhập để comment"
- Khi muốn like → Hiện popup: "Đăng nhập để like"
- Login prompts redirect sang trang đăng ký

Key Points:

- ✓ Guest limitations được enforce ngay từ UI
- ✓ Tạo động lực để Guest đăng ký tài khoản
- ✓ Tuân thủ BR-01: Chỉ user đã xác thực mới được tương tác

1.2 Flow 2: User Registration → First Post

Mô tả: Luồng đăng ký và onboarding cho user mới, từ khi điền form đăng ký cho đến khi tạo bài viết đầu tiên.

Các bước chính:

- User điền form đăng ký (email, username, password)
- System validate và tạo account với email_verified = FALSE
- Gửi email xác thực → User click link verify
- Đánh dấu email_verified = TRUE
- Hiện Welcome modal + Tutorial tạo bài viết
- User cập nhật profile (Avatar, Bio, Social links)
- Mở Rich Text Editor để tạo bài viết đầu tiên

Alternative Flows:

- Email/Username trùng → Show error và suggest alternatives
- Password validation lỗi → Hiện thị requirements
- User chưa verify email → Gửi reminder sau 24h

1.3 Flow 3: Author → Create Post with Video

Mô tả: Luồng tạo bài viết với video đính kèm, bao gồm quá trình upload và encode video.

Các bước chính:

- Author click "Tạo bài viết" → Mở Rich Text Editor
- Điền tiêu đề, nội dung, category, tags (max 5)
- Click "Thêm video" → Request presigned MinIO URL
- Client upload trực tiếp lên MinIO (bypass server)
- Hiển thị progress bar real-time
- Confirm upload → Queue encoding job (status: processing)
- FFmpeg worker encode sang HLS (1080p, 720p, 480p, 360p)
- Poll video status mỗi 5s → Khi ready → Show preview
- Click "Xuất bản" → Validate BR → Publish post
- Queue job: Generate AI embedding

Error Handling:

- Upload fail → Retry hoặc Save draft không có video
- Encoding fail → Retry tối đa 3 lần (BR-04)
- Validation lỗi → Highlight errors và return to editor

1.4 Flow 4: User → Search & Bookmark

Mô tả: Hệ thống hỗ trợ 3 loại tìm kiếm: Keyword (full-text), Semantic (vector), và AI Chat (RAG).

Keyword Search:

- Full-text search với PostgreSQL
- Tìm bài viết có chứa chính xác từ khóa
- Nhanh nhưng phải match text chính xác

Semantic Search:

- Generate query embedding với Ollama
- Vector similarity search với pgvector
- Tìm bài viết có nghĩa tương tự (không nhất thiết match từ khóa)
- VD: "quản lý state" → Tìm thấy bài về "useState, Redux, Context API"

RAG (AI Chat):

- User hỏi: "Tôi đã viết gì về Đà Lạt?"
- Generate embedding → Retrieve top 5 posts liên quan
- Build context prompt → Call Ollama Llama 3
- Stream AI response (SSE) với citations

Bookmark:

- User click "Lưu" → Hiện modal chọn folder
- Tạo folder mới hoặc chọn có sẵn
- Save bookmark → Icon đổi thành "Đã lưu"

1.5 Flow 5: Admin → Content Moderation

Mô tả: Admin có đầy đủ quyền kiểm duyệt nội dung, quản lý users, và xử lý báo cáo vi phạm.

Các chức năng chính:

- Xem danh sách reports (báo cáo nội dung vi phạm)
- Review nội dung bị báo cáo → Quyết định: Dismiss, Delete, hoặc Ban user

- Xem comments đánh dấu review (có từ ngữ nhạy cảm)
- Approve hoặc Delete comments spam
- Quản lý users: Xem activity logs, Block, hoặc Soft delete

Audit Trail:

- Mọi hành động của Admin được log vào activity_logs table
- Metadata: Lý do, reporter count, original author
- Gửi thông báo cho user bị ảnh hưởng
- Update dashboard stats sau mỗi action

Business Rules:

- Admin có toàn quyền với mọi nội dung (BR-01)
- Mọi action phải có audit trail để trách nhiệm giải trình

Tổng kết PHẦN A

5 Flow Diagrams đã cover:

- ✓ Guest journey (không login)
- ✓ Registration và onboarding
- ✓ Core workflow: tạo post + video
- ✓ Search với 3 modes (fulltext, semantic, RAG)
- ✓ Admin moderation

Các điểm cần chú ý:

1. Guest limitations được hiện rõ qua login prompts
2. Video processing có error handling (retry, failed states)
3. Search phân biệt 3 loại: keyword, semantic, AI chat
4. Admin có đầy đủ CRUD + logging

PHẦN B: USE CASES CHI TIẾT

Phần này document chi tiết các Use Cases quan trọng nhất theo format chuẩn. Mỗi UC bao gồm: Actor, Precondition, Main Flow, Alternative Flows, Business Rules applied, và Postcondition. Tất cả đều có SQL queries và code snippets cụ thể.

2.1 UC-01: ĐĂNG KÝ TÀI KHOẢN

UC-01.1: Đăng ký tài khoản (Email + Password)

Actor: Guest

Precondition:

- User chưa có tài khoản
- Truy cập trang đăng ký

Main Flow:

1. User click button "Đăng ký"
2. System hiển thị form với các trường: Email, Username, Password, Confirm Password
3. User điền thông tin và submit
4. System validate input (Zod schema)
5. System kiểm tra email/username đã tồn tại chưa
6. System hash password bằng bcrypt (cost factor 12)
7. System tạo record trong bảng users với email_verified = FALSE
8. System generate verification token (JWT, expire 24h)
9. System gửi email xác thực
10. System hiển thị: "Kiểm tra email để xác thực tài khoản"
11. Redirect về trang login

Alternative Flows:

A1: Email đã tồn tại (Step 5)

- System query: `SELECT * FROM users WHERE email = $1`
- Nếu tìm thấy → return error 409: "Email đã được sử dụng"
- Hiển thị error message phía dưới trường email
- Return to Step 3

A2: Username đã tồn tại (Step 5)

- Suggest alternatives: username1, username_2025, etc.
- Return to Step 3

A3: Validation lỗi (Step 4)

- Email invalid → "Email không hợp lệ"
- Password yếu → "Mật khẩu cần tối thiểu 8 ký tự, có chữ hoa, số, ký tự đặc biệt"
- Confirm password không khớp → Error message

Business Rules Applied:

- BR-01: User phải xác thực email mới được tạo bài viết
- Password policy: min 8 chars, uppercase, number, special char

Postcondition:

- Record user mới được tạo với email_verified = FALSE

- Email xác thực được gửi
- User có thể login nhưng bị giới hạn tính năng cho đến khi verify email

2.2 UC-02: TẠO BÀI VIẾT MỚI

UC-02.1: Tạo bài viết với Rich Text Editor

Actor: Registered User (email verified)

Precondition:

- User đã đăng nhập
- email_verified = TRUE
- Số bài viết của user < 100 (BR-02)

Main Flow:

1. User click "Tạo bài viết"
2. System kiểm tra: `SELECT COUNT(*) FROM posts WHERE author_id = $userId`
3. Nếu count < 100 → cho phép tạo
4. System hiển thị Rich Text Editor với các trường:
 - Title (required, 10-200 chars)
 - Content (required, min 50 chars) - TinyMCE/Tiptap
 - Category (required, dropdown)
 - Tags (optional, max 5 tags)
 - Featured Image (optional, max 5MB)
 - Video (optional, button "Upload Video")
5. User nhập tiêu đề
6. System auto-generate slug: `slugify(title) // "Hello World" → "hello-world"`
7. System kiểm tra slug uniqueness trong scope của user
8. Nếu trùng → append số: `hello-world-2, hello-world-3`
9. User nhập nội dung rich text
10. User chọn category và thêm tags
11. User click "Lưu nháp" hoặc "Xuất bản"
12. System validate theo BR-02
13. System tạo post record và quan hệ với category, tags
14. System tạo/update tags với usage_count
15. Hiển thị thông báo success
16. Redirect về trang chi tiết bài viết

Alternative Flows:

A1: Đạt giới hạn 100 bài viết (Step 3)

- count = 100 → return error 403
- Modal: "Bạn đã đạt giới hạn 100 bài viết. Vui lòng xóa bài viết cũ."
- End use case

A2: Validation lỗi (Step 12)

- Title < 10 chars → "Tiêu đề quá ngắn"
- Content < 50 chars → "Nội dung quá ngắn"
- Tags > 5 → "Tối đa 5 tags"
- Return to editor với highlight errors

A3: Upload featured image

- Kiểm tra file type (jpg, png, webp) và size (max 5MB)
- Upload trực tiếp lên MinIO
- Lưu presigned URL vào featured_image_url

A4: Auto-save draft (Background)

- System auto-save mỗi 30 giây

- Hiện thị "Đã lưu tự động" ở góc editor

Business Rules Applied:

- BR-01: Chỉ user đã verify email mới được tạo
- BR-02: Max 100 bài viết/user, max 5 tags
- BR-03: Tiêu đề phải unique trong scope của user

Postcondition:

- Bài viết mới với status = 'draft' hoặc 'published'
- Quan hệ với category, tags được thiết lập
- Nếu published → trigger embedding generation

2.3 UC-03: UPLOAD VÀ ENCODE VIDEO

UC-03.1: Upload video với presigned URL

Actor: Author (đang tạo/edit bài viết)

Precondition:

- User đang trong editor
- Bài viết chưa có video (max 1 video/post - BR-02)

Main Flow:

1. User click "Thêm video"
2. System hiển thị file picker
3. User chọn file video
4. System validate: MP4/MOV/AVI, max 2GB, max 30 phút
5. Frontend gửi: POST /api/videos/upload-url
6. Backend tạo video record với status = 'uploading'
7. Backend generate presigned MinIO URL (expire 1h)
8. Backend return: videoid, uploadUrl, expiresAt
9. Frontend upload trực tiếp lên MinIO với progress tracking
10. Hiển thị progress bar real-time
11. Upload xong → POST /api/videos/{videoid}/confirm
12. Backend update status = 'processing'
13. Queue encoding job vào BullMQ
14. UI hiển thị: "Video đang được xử lý..."
15. Frontend poll status mỗi 5 giây
16. Worker FFmpeg bắt đầu encode (UC-03.2)
17. Khi status = 'ready' → Hiển thị preview

Alternative Flows:

A1: File validation fail (Step 4)

- Wrong type → "Chỉ hỗ trợ MP4, MOV, AVI, MKV"
- File > 2GB → "Dung lượng tối đa 2GB"
- Duration > 30min → "Độ dài tối đa 30 phút"

A2: Upload fail - Network error (Step 9)

- Axios throws error (timeout, connection lost)
- Modal: "Upload thất bại. Thử lại?"
- User retry → return to Step 9
- User cancel → delete video record

A3: Presigned URL expired

- MinIO returns 403
- System auto-request new URL
- Retry upload (max 3 attempts)

Business Rules:

- BR-02: Max 1 video/post, 30 min, 2GB
- BR-04: Video chỉ hiển thị khi status = 'ready'

UC-03.2: Encode video với FFmpeg Worker

Actor: System (Background Worker)

Main Flow:

1. Worker nhận job từ BullMQ queue
2. Download raw video từ MinIO về /tmp
3. Extract metadata với FFprobe (duration, resolution, codec)
4. Update database với metadata
5. Generate thumbnail (frame @ 2s)
6. Upload thumbnail lên MinIO
7. Encode video sang HLS với 4 qualities:
 - 1080p: 5000k bitrate
 - 720p: 2800k bitrate
 - 480p: 1400k bitrate
 - 360p: 800k bitrate
8. Upload encoded files (.m3u8 + .ts segments)
9. Create master playlist
10. Update status = 'ready'
11. Delete raw video file (BR-04)
12. Cleanup temp files
13. Emit "video:ready" event via WebSocket

Error Handling:

A1: FFmpeg encoding error

- Parse error log
- Increment retry_count
- If retry_count < 3 → Re-queue với delay 5 phút
- If retry_count = 3 → status = 'failed' (BR-04)

A2: Timeout (encoding > 30 min)

- Kill FFmpeg process
- Status = 'failed', error = 'Encoding timeout'

Business Rules:

- BR-04: Processing fail > 3 lần → 'failed' vĩnh viễn
- BR-04: Auto-delete raw file sau encode thành công

2.4 UC-04: TƯƠNG TÁC (ENGAGEMENT)

UC-04.1: Bình luận trên bài viết

Actor: Registered User

Main Flow:

1. User xem bài viết → Scroll xuống section "Bình luận"
2. System hiển thị comments hiện có + comment box
3. User nhập nội dung (1-500 ký tự)
4. System validate real-time
5. User click "Gửi"
6. System check sensitive words
7. Tạo comment với is_flagged = hasSensitiveWords
8. Increment post.comment_count
9. Thông báo cho author
10. Frontend hiển thị comment ngay (optimistic update)
11. Nếu flagged → Badge "Đang chờ kiểm duyệt"

Alternative: Reply to comment (1-level)

- Click "Trả lời" → Set parent_id
- Validate: Không cho reply-to-reply

Alternative: Spam detection

- Check Redis: INCR comment:userId:count:today
- If > 50/day → Error 429: "Đạt giới hạn"

UC-04.2: Like/Unlike Toggle

Main Flow:

1. User click button "♥ Thích"
2. POST /api/posts/{postId}/like
3. Backend kiểm tra: SELECT FROM likes WHERE user_id AND post_id
4. Case 1: Chưa like → INSERT + Increment like_count
5. Case 2: Đã like → DELETE + Decrement like_count
6. Return: { liked: true/false, likeCount: 42 }
7. Frontend toggle UI: "♥ Đã thích" ↔ "□ Thích"
8. Gửi notification cho author (chỉ khi like)

Error Handling:

- Race condition → Unique constraint prevents duplicates
- Network error → Rollback optimistic update

UC-04.3: Bookmark với Folders

Main Flow:

1. User click icon "□ Lưu"
2. Modal: "Lưu vào thư mục"
3. Hiển thị folders: "Đọc sau", "Công việc", "Du lịch"
4. User chọn folder hoặc tạo mới
5. POST /api/bookmarks { postId, folderId }
6. Backend check duplicate
7. Insert bookmark hoặc update folder
8. Update folder.post_count
9. Toast: "□ Đã lưu vào 'Du lịch'"

10. Icon → "🔖 Đã lưu"

Alternative: Quick save

- Auto-save vào folder default "Đọc sau"
- Tạo folder nếu chưa có


Alternative: Unbookmark

- User click lại → Confirm "Xóa khỏi thư mục?"
- DELETE bookmark + update folder count

UC-04.4: Kiểm duyệt comment (Author)

Actor: Author

Main Flow:

1. Author nhận notification: "1 bình luận cần kiểm duyệt"
2. Click notification → Redirect đến bài viết
3. Hiển thị comment với badge [ Cần kiểm duyệt]
4. Author review
5. Case 1: Chấp nhận → Update is_flagged = FALSE
6. Case 2: Xóa → Soft delete + Decrement count
7. Case 3: Báo cáo spam → Delete + Increment user.spam_score
8. If spam_score >= 5 → Auto-block user

Auto-approve:

- Cronjob: Auto-approve comments pending > 24h

2.5 UC-05: TÌM KIẾM VỚI AI

UC-05.2: Semantic Search với Vector Embeddings

Actor: Any User (Guest hoặc Registered)

Main Flow:

1. User nhập query: "cách quản lý state trong React hooks"
2. Chọn mode "Tìm kiếm thông minh"
3. POST /api/search/semantic { query, limit: 10, minSimilarity: 0.7 }
4. Backend generate query embedding:
 - Call Ollama: POST http://ollama:11434/api/embeddings
 - Model: nomic-embed-text
 - Return: embedding float[1536]
5. Vector similarity search:
SELECT id, title, excerpt, 1 - (embedding <=> \$queryEmbedding) AS similarity
FROM posts
WHERE status = 'published' AND similarity >= 0.7
ORDER BY similarity DESC
LIMIT 10
6. Return results với similarity scores
7. Frontend hiển thị kết quả ranked by similarity
8. User click vào post để đọc

Alternative Flows:

A1: Không tìm thấy (similarity < 0.7)

- Retry với minSimilarity = 0.5
- Nếu vẫn empty → Suggest: "Thử từ khóa đơn giản hơn"
- Show related tags

A2: Ollama service down

- Fallback sang Full-text search
- Log error
- Note: "Tìm kiếm cơ bản (AI tạm không khả dụng)"

A3: Cache hit

- Check Redis: GET search:semantic:hash(query)
- If exists → Return cached (TTL 5 phút)
- Skip Steps 4-5

UC-05.3: RAG - Hỏi đáp với AI

Actor: Registered User

Main Flow:

1. User click "Hỏi AI"
2. User nhập: "Tôi đã viết gì về Đà Lạt tháng trước?"
3. POST /api/search/rag { question, userId, stream: true }
4. Generate question embedding (như UC-05.2)
5. Retrieve top 5 relevant posts:
SELECT title, content, published_at
WHERE author_id = \$userId
ORDER BY embedding <=> \$questionEmbedding
LIMIT 5

6. Build context prompt:
"Bạn là trợ lý AI. Dựa trên các bài viết:
[Post 1] Tiêu đề: ..., Ngày: ..., Nội dung: ...
[Post 2] ...
Hãy trả lời: [question]"
7. Stream response từ Ollama:
POST `http://ollama:11434/api/generate`
{ model: 'llama3', prompt, stream: true }
8. Frontend hiển thị real-time (typewriter effect)
9. Khi done → Show "Nguồn tham khảo" với clickable links

Alternative:

- A1: Không tìm thấy posts
- Return: "Xin lỗi, không tìm thấy bài viết liên quan"
 - Suggest: "Thử hỏi về chủ đề khác"

- A2: Ollama timeout
- Stream bị interrupt sau 30s
 - Error: "AI đang quá tải"
 - Fallback: Hiển thị list posts liên quan

Business Rules:

- BR-05: Embedding auto-generate khi publish
- BR-05: Cache results 5 phút

2.6 UC-06: THỐNG KÊ & ANALYTICS

UC-06.1: Dashboard tác giả

Actor: Author

Main Flow:

1. Author click "Thống kê"
2. Redirect /dashboard/analytics
3. Backend query aggregated data

Query 1: Tổng quan

```
SELECT COUNT(*) as total_posts,  
SUM(view_count) as total_views,  
SUM(like_count) as total_likes,  
SUM(comment_count) as total_comments  
FROM posts WHERE author_id = $userId
```

Query 2: Views trend (30 ngày)

```
SELECT DATE(viewed_at) as date,  
COUNT(DISTINCT session_id) as unique_views  
FROM post_views  
WHERE post_id IN (SELECT id FROM posts WHERE author_id = $userId)  
AND viewed_at >= NOW() - INTERVAL '30 days'  
GROUP BY DATE(viewed_at)
```

Query 3: Top posts

```
SELECT title, view_count, like_count  
FROM posts WHERE author_id = $userId  
ORDER BY view_count DESC LIMIT 10
```

Query 4: Traffic sources

```
SELECT referrer, COUNT(DISTINCT session_id) as visitors  
FROM post_views  
WHERE post_id IN (...)  
GROUP BY referrer
```

4. Frontend render dashboard:

- Overview cards (posts, views, likes, comments)
 - Line chart: Views trend 30 ngày (Recharts)
 - Table: Top 10 bài viết
 - Pie chart: Traffic sources
5. Author có thể filter by date (7d, 30d, 90d, custom)
6. Export data to CSV

Alternative:

A1: No data yet

- Empty state: "Chưa có dữ liệu thống kê"
- Button: [Tạo bài viết mới]

A2: Export CSV

- Generate CSV: Title, Views, Likes, Comments, Date
- Download file

UC-06.2: View Tracking (Session-based)

Actor: System (Background)

Main Flow:

1. User visit /posts/{slug}
2. Frontend generate/retrieve sessionId:
`sessionId = localStorage.getItem('sessionId') || uuid()`
3. Send tracking beacon:
`POST /api/analytics/track-view`
`{ postId, sessionId, referrer, userAgent }`
4. Backend check unique view:
`SELECT id FROM post_views`
`WHERE post_id = $postId AND session_id = $sessionId`
`AND DATE(viewed_at) = CURRENT_DATE`
5. Case 1: Chưa view hôm nay → `INSERT` + Increment view_count
6. Case 2: Đã view hôm nay → Skip (không tăng counter)
7. Return 204 No Content

Alternative:

A1: User đã login

- Include userId trong tracking
- Giúp track: User nào đọc bài nào

A2: Bot detection

- Check user agent: `/bot|crawler|spider/i.test(userAgent)`
- If bot → Log nhưng KHÔNG tăng view_count
- Insert với flag `is_bot = TRUE`

Business Rules:

- BR-06.2: View unique per session per day
- Session A view lúc 9h → +1
- Session A view lúc 11h (same day) → không tăng
- Session A view ngày hôm sau → +1

PHẦN C: BUSINESS RULES VALIDATION

Phần này tạo bảng kiểm tra xem Business Rule nào được áp dụng cho Use Case nào. Đảm bảo mọi BR đều được implement đúng chỗ và không bị miss khi coding.

3.1 UC × BR MATRIX

Use Case	BR-01	BR-02	BR-03	BR-04	BR-05
UC-01.1 Đăng ký	☑	-	-	-	-
UC-01.3 Verify email	☑	-	-	-	-
UC-02.1 Tạo bài viết	☑	☑	☑	-	-
UC-02.2 Chỉnh sửa	☑	-	☑	-	☑
UC-02.4 Xuất bản	☑	-	-	-	☑
UC-03.1 Upload video	☑	☑	-	-	-
UC-03.2 Encode video	-	-	-	☑	-
UC-04.1 Bình luận	☑	-	☑	-	-
UC-04.2 Like	☑	-	-	-	-
UC-04.3 Bookmark	☑	-	-	-	-
UC-04.4 Kiểm duyệt	☑	-	☑	-	-
UC-05.2 Semantic Search	-	-	-	-	☑
UC-05.3 RAG	-	-	-	-	☑
UC-06.1 Dashboard	-	-	-	-	-
UC-06.2 View Tracking	-	-	-	-	-

Chú thích: ☑ = BR được enforce | - = Không áp dụng

3.2 CHI TIẾT TỪNG BUSINESS RULE

BR-01: Xác thực & Phân quyền

- Chỉ user đã xác thực email mới được tạo bài viết
- User chỉ được sửa/xóa bài viết của chính mình
- Admin có toàn quyền với mọi nội dung

Enforcement:

- Middleware: requireVerifiedEmail - Check email_verified = TRUE
- Service: canEditPost() - Check author_id = currentUser.id OR isAdmin
- Applied in: UC-01.1, 01.3, 02.1, 02.2, 04.1-04.4

BR-02: Giới hạn Upload

- Mỗi user: Tối đa 100 bài viết

- Mỗi bài viết: Tối đa 1 video
- Video: Tối đa 30 phút, 2GB
- Ảnh thumbnail: Tối đa 5MB

Enforcement:

- Query: `SELECT COUNT(*) FROM posts WHERE author_id = $userId (before create)`
- Validate file.size, extract metadata duration
- Applied in: UC-02.1, 03.1

BR-03: Nội dung

- Tiêu đề bài viết phải unique trong phạm vi user
- Không cho phép nội dung spam (kiểm tra duplicate)
- Comment có từ ngữ nhạy cảm sẽ được đánh dấu review

Enforcement:

- `generateUniqueSlug()` - Check slug existence, append number if duplicate
- `checkSensitiveWords()` - Scan comment content
- Applied in: UC-02.1, 02.2, 04.1, 04.4

BR-04: Video Processing

- Video chỉ hiển thị khi trạng thái = 'ready'
- Nếu processing fail > 3 lần → status 'failed' vĩnh viễn
- Tự động xóa raw video sau encode thành công

Enforcement:

- Worker: Increment `retry_count` on error
- If `retry_count` >= 3 → `UPDATE status = 'failed'`
- After success: `DELETE raw file + UPDATE raw_file_path = NULL`
- Applied in: UC-03.2

BR-05: Search & AI

- Embedding auto-generate khi bài viết publish
- Re-generate embedding khi content thay đổi > 20%
- Cache kết quả search trong 5 phút

Enforcement:

- On publish: Queue embedding job
- On update: Calculate content diff, if > 20% → re-generate
- Redis cache: `SETEX search:hash TTL 300 seconds`
- Applied in: UC-02.2, 02.4, 05.2, 05.3

3.3 COVERAGE REPORT

BR	Description	Use Cases Applied	Coverage
BR-01	Xác thực & Phân quyền	8 UCs	☐ 100%
BR-02	Giới hạn Upload	2 UCs	☐ 100%
BR-03	Nội dung	4 UCs	☐ 100%
BR-04	Video Processing	1 UC	☐ 100%

BR-05	Search & AI	4 UCs	☐ 100%
-------	-------------	-------	--------

Kết luận: ☐ Tất cả 5 Business Rules đều được enforce đầy đủ trong các Use Cases tương ứng.

3.4 IMPLEMENTATION CHECKLIST

Khi coding, developer cần đảm bảo:

BR-01 Checklist:

- ☐ email_verified check trong middleware
- ☐ canEdit() check author ownership
- ☐ isAdmin bypass cho admin
- ☐ Unit tests cho permission logic
- ☐ Integration tests cho auth flows

BR-02 Checklist:

- ☐ Query count posts trước khi tạo
- ☐ Validate file size client-side và server-side
- ☐ Extract video metadata để check duration
- ☐ Unit tests cho validation functions
- ☐ E2E tests cho upload limits

BR-03 Checklist:

- ☐ generateUniqueSlug() với retry logic
- ☐ Sensitive words dictionary (DB table)
- ☐ Duplicate comment check (1-minute window)
- ☐ Auto-approve comments sau 24h (cronjob)
- ☐ Unit tests cho slug generation

BR-04 Checklist:

- ☐ Retry count increment trong worker
- ☐ Status transition: uploading → processing → ready/failed
- ☐ Delete raw file sau encode (cleanup job)
- ☐ Error logging cho failed encodes
- ☐ Integration tests cho encoding pipeline

BR-05 Checklist:

- ☐ Queue embedding job khi publish
- ☐ Content diff calculation (20% threshold)
- ☐ Redis cache với TTL 5 phút
- ☐ Cache invalidation khi post updated
- ☐ Unit tests cho caching logic

3.5 TRACEABILITY MATRIX

Requirement	Business Rule	Use Cases	Modules
REQ-001: User auth	BR-01	UC-01.1, 01.3	AuthService, Middleware
REQ-002: Create post	BR-01, BR-02, BR-03	UC-02.1	PostService

REQ-003: Upload video	BR-02, BR-04	UC-03.1, 03.2	VideoService, Worker
REQ-004: Comment system	BR-01, BR-03	UC-04.1, 04.4	CommentService
REQ-005: AI Search	BR-05	UC-05.2, 05.3	SearchService, OllamaClient

Lợi ích của Traceability Matrix:

- Dễ review: "BR-XX được enforce ở đâu?"
- Không miss BR khi coding
- Dễ viết test cases (đã có test IDs)
- Dễ onboard developer mới vào dự án