**TÀI LIỆU ĐẶC TẢ YÊU CẦU PHẦN MỀM**

**(Software Requirement Specification - SRS)**

**Hệ thống quản lý và đăng bài blog tự động - AutoBlog**

Phiên bản 4.0

**Được soạn bởi: Lê Xuân Bá Quốc**

**Nguyễn Gia Bảo**

**Nguyễn Đăng Trường**

**Nguyễn Lê Nhật Tuấn**

**Đà Nẵng, 6/2025**

**MỤC LỤC**

[CHƯƠNG 1. TÀI LIỆU XÁC ĐỊNH YÊU CẦU 11](#_Toc199871405)

[I. Đề cương dự án 11](#_Toc199871406)

[1. Các vấn đề chính cần giải quyết 11](#_Toc199871407)

[2. Các giải pháp cho từng vấn đề: 11](#_Toc199871408)

[II. Nghiên cứu khả thi 12](#_Toc199871409)

[1. Tính khả thi về kinh tế 12](#_Toc199871410)

[2. Tính khả thi về kỹ thuật 14](#_Toc199871411)

[III. Tài liệu yêu cầu 15](#_Toc199871412)

[1. Yêu cầu chức năng 15](#_Toc199871413)

[2. Yêu cầu phi chức năng 15](#_Toc199871414)

[IV. Danh sách rủi ro 16](#_Toc199871415)

[V. Kế hoạch ban đầu 21](#_Toc199871416)

[1. Giai đoạn 1: Xác định yêu cầu (Tuần 1) 21](#_Toc199871417)

[2. Giai đoạn 2: Phân tích (Tuần 2) 21](#_Toc199871418)

[3. Giai đoạn 3: Thiết kế hệ thống (Tuần 3) 21](#_Toc199871419)

[4. Giai đoạn 4: Phát triển tính năng cốt lõi (Tuần 4 - 6) 22](#_Toc199871420)

[5. Giai đoạn 5: Tích hợp và kiểm thử (Tuần 7 - 8) 22](#_Toc199871421)

[6. Giai đoạn 6: Triển khai (Tuần 9) 23](#_Toc199871422)

[7. Giai đoạn 7: Đánh giá và hoàn thiện (Tuần 10) 23](#_Toc199871423)

[VI. Đề xuất 23](#_Toc199871424)

[1. Ước tính thời gian: 24](#_Toc199871425)

[2. Dự toán chi phí 24](#_Toc199871426)

[3. Môi trường triển khai 25](#_Toc199871427)

[CHƯƠNG 2. TÀI LIỆU PHÂN TÍCH YÊU CẦU 26](#_Toc199871428)

[I. Đặc tả chức năng 26](#_Toc199871429)

[1. Tổng quan về hệ thống 26](#_Toc199871430)

[2. Yêu cầu chức năng 26](#_Toc199871431)

[3. Trường hợp sử dụng 26](#_Toc199871432)

[4. Tiêu chí chấp nhận 26](#_Toc199871433)

[II. Kế hoạch dự án cuối cùng 27](#_Toc199871434)

[1. Phạm vi dự án 27](#_Toc199871435)

[2. Các mốc quan trọng (Milestones) chi tiết 29](#_Toc199871436)

[3. Dòng thời gian và phân bổ nguồn lực 30](#_Toc199871437)

[III. Tài liệu đề xuất phát triển 32](#_Toc199871438)

[1. Tóm tắt điều hành 32](#_Toc199871439)

[2. Mục tiêu dự án 32](#_Toc199871440)

[3. Phương pháp luận 33](#_Toc199871441)

[IV. Thiết kế tổng thể 35](#_Toc199871442)

[1. Thiết kế kiến trúc 35](#_Toc199871443)

[2. Sơ đồ luồng dữ liệu tổng thể 36](#_Toc199871444)

[3. Điểm tích hợp 36](#_Toc199871445)

[CHƯƠNG 3. TÀI LIỆU ĐẶC TẢ THIẾT KẾ 41](#_Toc199871446)

[I. GIỚI THIỆU 41](#_Toc199871447)

[1. Tổng quan 41](#_Toc199871448)

[2. Phạm vi 41](#_Toc199871449)

[II. YÊU CẦU 43](#_Toc199871450)

[1. Yêu cầu chức năng 43](#_Toc199871451)

[2. Yêu cầu phi chức năng 44](#_Toc199871452)

[III. CÁC YẾU TỐ THIẾT KẾ 45](#_Toc199871453)

[1. Thiết kế cơ sở dữ liệu 45](#_Toc199871454)

[2. Thiết kế hướng đối tượng 46](#_Toc199871455)

[3. Thiết kế UI/UX 63](#_Toc199871456)

[IV. CÂN NHẮC TRIỂN KHAI 65](#_Toc199871457)

[1. Ngăn xếp công nghệ 65](#_Toc199871458)

[2. Tích hợp 66](#_Toc199871459)

[V. KIỂM TRA VÀ XÁC NHẬN 69](#_Toc199871460)

[1. Chiến lược thử nghiệm 69](#_Toc199871461)

[2. Tiêu chí xác nhận 71](#_Toc199871462)

[CHƯƠNG 4. TÀI LIỆU THỰC HIỆN 73](#_Toc199871463)

[US01 – Đăng ký/Đăng nhập 73](#_Toc199871464)

[1.1. Mô tả User Story 73](#_Toc199871465)

[1.2. Thiết kế chức năng theo User Story 73](#_Toc199871466)

[1.3. Tài liệu sử dụng (User Guide) 74](#_Toc199871467)

[US02 – Tạo bài viết mới 75](#_Toc199871468)

[1.1. Mô tả User Story 75](#_Toc199871469)

[1.2. Thiết kế chức năng theo Story 76](#_Toc199871470)

[1.3. Tài liệu sử dụng (User Guide) 76](#_Toc199871471)

[US03 – Tự động tạo nội dung AI 77](#_Toc199871472)

[1.1. Mô tả User Story 77](#_Toc199871473)

[1.2. Thiết kế chức năng theo Story 78](#_Toc199871474)

[1.3. Tài liệu sử dụng (User Guide) 79](#_Toc199871475)

[US04 – Chỉnh sửa bài viết 80](#_Toc199871476)

[1.1. Mô tả User Story 80](#_Toc199871477)

[1.2. Thiết kế chức năng theo Story 80](#_Toc199871478)

[1.3. Tài liệu sử dụng (User Guide) 81](#_Toc199871479)

[US05 – Xóa bài viết 82](#_Toc199871480)

[1.1. Mô tả User Story 82](#_Toc199871481)

[1.2. Thiết kế chức năng theo Story 83](#_Toc199871482)

[1.3. Tài liệu sử dụng (User Guide) 83](#_Toc199871483)

[US06 – Phân loại bài viết theo danh mục 84](#_Toc199871484)

[1.1. Mô tả User Story 84](#_Toc199871485)

[1.2. Thiết kế chức năng theo Story 85](#_Toc199871486)

[1.3. Tài liệu sử dụng (User Guide) 85](#_Toc199871487)

[US07 – Lọc và sắp xếp bài viết 86](#_Toc199871488)

[1.1. Mô tả User Story 86](#_Toc199871489)

[1.2. Thiết kế chức năng theo Story 87](#_Toc199871490)

[1.3. Tài liệu sử dụng (User Guide) 88](#_Toc199871491)

[US08 – Lên lịch đăng bài tự động 88](#_Toc199871492)

[1.1. Mô tả User Story 88](#_Toc199871493)

[1.2. Thiết kế chức năng theo Story 89](#_Toc199871494)

[1.3. Tài liệu sử dụng (User Guide) 90](#_Toc199871495)

[US09 – Quản lý hình ảnh và tệp đính kèm 90](#_Toc199871496)

[1.1. Mô tả User Story 90](#_Toc199871497)

[1.2. Thiết kế chức năng theo Story 91](#_Toc199871498)

[1.3. Tài liệu sử dụng (User Guide) 91](#_Toc199871499)

[US10 – Tìm kiếm bài viết 92](#_Toc199871500)

[1.1. Mô tả User Story 92](#_Toc199871501)

[1.2. Thiết kế chức năng theo Story 92](#_Toc199871502)

[1.3. Tài liệu sử dụng (User Guide) 93](#_Toc199871503)

[US11 – Tối ưu hóa SEO tự động 94](#_Toc199871504)

[1.1. Mô tả User Story 94](#_Toc199871505)

[1.2. Thiết kế chức năng theo Story 94](#_Toc199871506)

[1.3. Tài liệu sử dụng (User Guide) 95](#_Toc199871507)

[US12 – Thông báo khi đăng bài thành công 96](#_Toc199871508)

[1.1. Mô tả User Story 96](#_Toc199871509)

[1.2. Thiết kế chức năng theo Story 97](#_Toc199871510)

[1.3. Tài liệu sử dụng (User Guide) 97](#_Toc199871511)

[US13 – Thống kê hiệu suất bài viết 98](#_Toc199871512)

[1.1. Mô tả User Story 98](#_Toc199871513)

[1.2. Thiết kế chức năng theo Story 99](#_Toc199871514)

[1.3. Tài liệu sử dụng (User Guide) 99](#_Toc199871515)

[US14 – Chia sẻ bài viết lên mạng xã hội 100](#_Toc199871516)

[1.1. Mô tả User Story 100](#_Toc199871517)

[1.2. Thiết kế chức năng theo Story 101](#_Toc199871518)

[1.3. Tài liệu sử dụng (User Guide) 101](#_Toc199871519)

[US15 – Giao diện thân thiện đa thiết bị 103](#_Toc199871520)

[1.1. Mô tả User Story 103](#_Toc199871521)

[1.2. Thiết kế chức năng theo Story 103](#_Toc199871522)

[1.3. Tài liệu sử dụng (User Guide) 104](#_Toc199871523)

[US16 – Chế độ xem trước bài viết 106](#_Toc199871524)

[1.1. Mô tả User Story 106](#_Toc199871525)

[1.2. Thiết kế chức năng theo Story 107](#_Toc199871526)

[1.3. Tài liệu sử dụng (User Guide) 107](#_Toc199871527)

[US17 – Nhập/Xuất dữ liệu 108](#_Toc199871528)

[1.1. Mô tả User Story 108](#_Toc199871529)

[1.2. Thiết kế chức năng theo Story 109](#_Toc199871530)

[1.3. Tài liệu sử dụng (User Guide) 109](#_Toc199871531)

[US18 – Bảo mật hệ thống 110](#_Toc199871532)

[1.1. Mô tả User Story 110](#_Toc199871533)

[1.2. Thiết kế chức năng theo Story 111](#_Toc199871534)

[1.3. Tài liệu sử dụng (User Guide) 111](#_Toc199871535)

[US19 – Sao lưu tự động 112](#_Toc199871536)

[1.1. Mô tả User Story 112](#_Toc199871537)

[1.2. Thiết kế chức năng theo Story 113](#_Toc199871538)

[1.3. Tài liệu sử dụng (User Guide) 113](#_Toc199871539)

[US20 – Tương tác với người đọc 114](#_Toc199871540)

[1.1. Mô tả User Story 114](#_Toc199871541)

[1.2. Thiết kế chức năng theo Story 115](#_Toc199871542)

[1.3. Tài liệu sử dụng (User Guide) 115](#_Toc199871543)

[US21 – Tùy chỉnh giao diện blog 116](#_Toc199871544)

[1.1. Mô tả User Story 116](#_Toc199871545)

[1.2. Thiết kế chức năng theo Story 117](#_Toc199871546)

[1.3. Tài liệu sử dụng (User Guide) 118](#_Toc199871547)

[US22 – Đề xuất nội dung liên quan 118](#_Toc199871548)

[1.1. Mô tả User Story 118](#_Toc199871549)

[1.2. Thiết kế chức năng theo Story 119](#_Toc199871550)

[1.3. Tài liệu sử dụng (User Guide) 119](#_Toc199871551)

[CHƯƠNG 5. TÀI LIỆU KIỂM THỬ 121](#_Toc199871552)

[I. Sprint 1 121](#_Toc199871553)

[1. Kế hoạch kiểm thử 121](#_Toc199871554)

[2. Test Cases 122](#_Toc199871555)

[3. Báo cáo bug 126](#_Toc199871556)

[4. Báo cáo tổng quan kết quả kiểm thử 126](#_Toc199871557)

[II. Sprint 2 126](#_Toc199871558)

[1. Kế hoạch kiểm thử 126](#_Toc199871559)

[2. Test Cases 127](#_Toc199871560)

[3. Báo cáo bug 136](#_Toc199871561)

[4. Báo cáo tổng quan kết quả kiểm thử 136](#_Toc199871562)

[III. Sprint 3 137](#_Toc199871563)

[1. Kế hoạch kiểm thử 137](#_Toc199871564)

[2. Test Cases 138](#_Toc199871565)

[3. Báo cáo bug 144](#_Toc199871566)

[4. Báo cáo tổng quan kết quả kiểm thử 144](#_Toc199871567)

[IV. Sprint 4 145](#_Toc199871568)

[1. Kết quả kiểm thử 145](#_Toc199871569)

[2. Test Cases 146](#_Toc199871570)

[CHƯƠNG 6. TÀI LIỆU CHẤP NHẬN (UAT) 151](#_Toc199871571)

[I. Sprint 1: Nền tảng cơ sở 151](#_Toc199871572)

[1. Mục đích 151](#_Toc199871573)

[2. Phạm vi Thử nghiệm Chấp nhận 151](#_Toc199871574)

[3. Môi trường Thử nghiệm Chấp nhận 151](#_Toc199871575)

[4. Quy trình Tổng quan Thực hiện UAT 152](#_Toc199871576)

[5. Vai trò và Trách nhiệm 152](#_Toc199871577)

[6. Tiêu chí Chấp nhận (Pass/Fail Criteria) 152](#_Toc199871578)

[7. Quy trình Ra Quyết định Chấp nhận 153](#_Toc199871579)

[8. Hình thức Xác nhận Chấp nhận 153](#_Toc199871580)

[II. Sprint 2: Chức năng quản lý nội dung cơ bản 154](#_Toc199871581)

[1. Mục đích 154](#_Toc199871582)

[2. Phạm vi thử nghiệm chấp nhận 154](#_Toc199871583)

[3. Môi trường thử nghiệm chấp nhận 154](#_Toc199871584)

[4. Quy trình tổng quan thực hiện UAT 155](#_Toc199871585)

[5. Vai trò và trách nhiệm 155](#_Toc199871586)

[6. Tiêu chí chấp nhận (Pass/Fail Criteria) 156](#_Toc199871587)

[7. Quy trình ra quyết định chấp nhận 156](#_Toc199871588)

[8. Hình thức xác nhận chấp nhận 157](#_Toc199871589)

[III. Sprint 3: Trải nghiệm người dùng nâng cao và AI 157](#_Toc199871590)

[1. Mục đích 157](#_Toc199871591)

[2. Phạm vi Thử nghiệm chấp nhận 157](#_Toc199871592)

[3. Môi trường thử nghiệm chấp nhận 157](#_Toc199871593)

[4. Quy trình tổng quan thực hiện UAT 158](#_Toc199871594)

[5. Vai trò và Trách nhiệm 158](#_Toc199871595)

[6. Tiêu chí Chấp nhận (Pass/Fail Criteria) 159](#_Toc199871596)

[7. Quy trình ra quyết định chấp nhận 159](#_Toc199871597)

[8. Hình thức xác nhận chấp nhận 159](#_Toc199871598)

[IV. Sprint 4: SEO và Lên Lịch Tự Động 160](#_Toc199871599)

[1. Mục đích 160](#_Toc199871600)

[2. Phạm vi thử nghiệm chấp nhận 160](#_Toc199871601)

[3. Môi trường thử nghiệm chấp nhận 160](#_Toc199871602)

[4. Quy trình tổng quan thực hiện UAT 161](#_Toc199871603)

[5. Vai trò và trách nhiệm 161](#_Toc199871604)

[6. Tiêu chí chấp nhận (Pass/Fail Criteria) 161](#_Toc199871605)

[7. Quy trình ra quyết định chấp nhận 162](#_Toc199871606)

[8. Hình thức xác nhận chấp nhận 162](#_Toc199871607)

**DANH MỤC HÌNH ẢNH**

[Hình 1.Biểu đồ chi phí 13](#_Toc199871124)

[Hình 2. Biểu đồ Gantt 25](#_Toc199871125)

[Hình 3. Sơ đồ thứ tự công việc 31](#_Toc199871126)

[Hình 4. Danh sách yêu cầu nguồn lực 32](#_Toc199871127)

[Hình 5. Thiết kế kiến trúc 36](#_Toc199871128)

[Hình 6. Sơ đồ luồng dữ liệu tổng thể 37](#_Toc199871129)

[Hình 7. Sơ đồ ERD 46](#_Toc199871130)

[Hình 8. Sơ đồ usecase tổng quát 47](#_Toc199871131)

[Hình 9. Sơ đồ usecase quản lý danh mục 48](#_Toc199871132)

[Hình 10. Sơ đồ usecase quản lý tài khoản 50](#_Toc199871133)

[Hình 11. Sơ dồ usecase tìm kiếm 52](#_Toc199871134)

[Hình 12. Sơ đồ usecase quản lý bài viết 54](#_Toc199871135)

[Hình 13. Sơ đồ usecase xem bài viết 57](#_Toc199871136)

[Hình 14. Sơ đồ lớp 59](#_Toc199871137)

[Hình 15. Sơ đồ hoạt động tạo bài viết 59](#_Toc199871138)

[Hình 16. Sơ đồ hoạt động tìm kiếm 60](#_Toc199871139)

[Hình 17. Sơ đồ hoạt động cập nhật bài viết 60](#_Toc199871140)

[Hình 18. Sơ đồ hoạt động xoá bài viết 60](#_Toc199871141)

[Hình 19. Sơ đồ tuần tự tạo bài viết 61](#_Toc199871142)

[Hình 20. Sơ đồ tuần tự tìm kiếm 62](#_Toc199871143)

[Hình 21. Sơ đồ tuần tự cập nhật bài viết 63](#_Toc199871144)

[Hình 22. Sơ đồ tuần tự xoá bài viết 64](#_Toc199871145)

[Hình 23. Đăng nhập 75](#_Toc199871146)

[Hình 24. Đăng ký 76](#_Toc199871147)

[Hình 25. Tạo bài viết 78](#_Toc199871148)

[Hình 26. AI tạo nội dung bài viết 80](#_Toc199871149)

[Hình 27. Cập nhật bài viết 83](#_Toc199871150)

[Hình 28. Xoá bài viết 85](#_Toc199871151)

[Hình 29. Phân loại bài viết theo danh mục 87](#_Toc199871152)

[Hình 30. Lọc và sắp xếp bài viết 89](#_Toc199871153)

[Hình 31. Tìm kiếm bài viết 94](#_Toc199871154)

[Hình 32. Thông báo đăng bài thành công 99](#_Toc199871155)

[Hình 33. Chia sẻ bài viết lên mạng xã hội 103](#_Toc199871156)

[Hình 34. Giao diện trên thiết bị mobile 106](#_Toc199871157)

[Hình 35. Giao diện trên desktop 107](#_Toc199871158)

[Hình 36. Chế độ xem trước bài viết 109](#_Toc199871159)

[Hình 37. Tương tác với người đọc 117](#_Toc199871160)

**DANH MỤC BẢNG**

[Bảng 1. Bảng phân tích tính khả thi về kỹ thuật 14](#_Toc199871608)

[Bảng 2. Yêu cầu chức năng 15](#_Toc199871609)

[Bảng 3. Yêu cầu phi chức năng 15](#_Toc199871610)

[Bảng 4. Bảng danh sách rủi ro 16](#_Toc199871611)

[Bảng 5. Mục tiêu dự án 32](#_Toc199871612)

[Bảng 6. Công nghệ sử dụng 33](#_Toc199871613)

[Bảng 7. Ước tính ngân sách 34](#_Toc199871614)

[Bảng 8. Yêu cầu chức năng 43](#_Toc199871615)

[Bảng 9. Yêu cầu phi chức năng 44](#_Toc199871616)

[Bảng 10. Đặc tả usecase quản lý danh mục 47](#_Toc199871617)

[Bảng 11. Đặc tả usecase quản lý tài khoản 49](#_Toc199871618)

[Bảng 12. Đặc tả usecase tìm kiếm 51](#_Toc199871619)

[Bảng 13. Đặc tả usecase quản lý bài viết 53](#_Toc199871620)

[Bảng 14. Đặc tả usecase xem bài viết 56](#_Toc199871621)

[Bảng 15. Tiêu chí xác nhận yêu cầu chức năng 71](#_Toc199871622)

[Bảng 16. Tiêu chí xác nhận yêu cầu phi chức năng 72](#_Toc199871623)

[Bảng 17. Bảng các test case của sprint 1 122](#_Toc199871624)

[Bảng 18. Bảng các bug của sprint 1 126](#_Toc199871625)

[Bảng 19. Bảng các test case của sprint 2 127](#_Toc199871626)

[Bảng 20. Bảng các bug của sprint 2 136](#_Toc199871627)

[Bảng 21. Bảng các test case của sprint 3 138](#_Toc199871628)

[Bảng 22. Bảng các bug của sprint 3 144](#_Toc199871629)

[Bảng 23. Bảng các test case của sprint 4 146](#_Toc199871630)

# CHƯƠNG 1. TÀI LIỆU XÁC ĐỊNH YÊU CẦU

## I. Đề cương dự án

### 1. Các vấn đề chính cần giải quyết

* Việc quản lý nội dung blog thủ công tốn nhiều thời gian và công sức.
* Không có hệ thống tự động đăng bài theo lịch trình, gây gián đoạn nội dung.
* Thiếu khả năng tích hợp với các nền tảng bên ngoài như mạng xã hội.

### 2. Các giải pháp cho từng vấn đề:

**Vấn đề: Việc quản lý nội dung blog thủ công tốn nhiều thời gian và công sức**

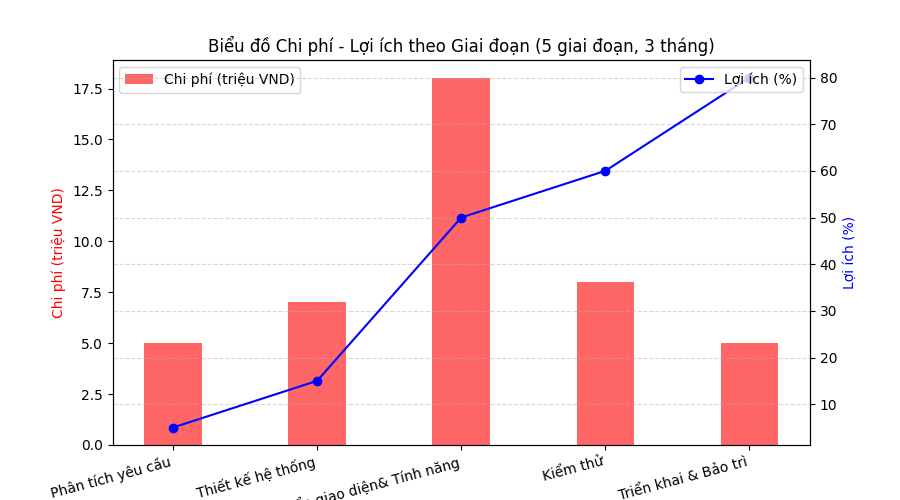
* **Giải pháp:** Sử dụng AI để tự động tạo bài viết dựa trên chủ đề, từ khóa và phong cách nội dung mong muốn. AI có thể phân tích dữ liệu, tổng hợp thông tin từ nhiều nguồn để tạo ra nội dung chất lượng cao mà không cần biên tập viên viết từ đầu.
* **Lợi ích:**
  + Tiết kiệm thời gian viết bài, giúp tập trung vào chiến lược nội dung.
  + Giảm công sức quản lý nội dung thủ công, tối ưu hóa quy trình làm việc.

**Vấn đề: Không có hệ thống tự động đăng bài theo lịch trình, gây gián đoạn nội dung**

* **Giải pháp**: Tích hợp hệ thống lập lịch tự động, cho phép AI tạo và đăng bài theo khung giờ tối ưu dựa trên phân tích hành vi người đọc.
* **Lợi ích:**
  + Duy trì luồng nội dung liên tục mà không cần can thiệp thủ công.
  + Tối ưu hóa thời điểm đăng bài để tiếp cận nhiều người dùng hơn.
  + Giúp doanh nghiệp/blogger duy trì sự chuyên nghiệp và ổn định trong việc cung cấp nội dung.

## II. Nghiên cứu khả thi

### 1. Tính khả thi về kinh tế



Hình 1.Biểu đồ chi phí

**Giai đoạn 1: Phân tích yêu cầu (Chi phí: 5 triệu VND, Lợi ích: 5%)**

* **Công việc chính:**
  + Nghiên cứu thị trường, xác định khách hàng mục tiêu.
  + Thu thập và phân tích yêu cầu để xác định phạm vi dự án.
  + Xây dựng tài liệu mô tả yêu cầu
* **Lợi ích:**
  + Giúp xác định đúng hướng phát triển sản phẩm.
  + Giảm rủi ro thay đổi yêu cầu sau này.
  + Lợi ích thấp vì chưa có sản phẩm cụ thể, nhưng có nền tảng cho các bước sau.

**Giai đoạn 2: Thiết kế hệ thống (Chi phí: 7 triệu VND, Lợi ích: 15%)**

* **Công việc chính:**
  + Thiết kế kiến trúc hệ thống, xác định công nghệ sử dụng.
  + Lên kế hoạch cơ sở dữ liệu, API, bảo mật.
  + Thiết kế UI/UX sơ bộ để đảm bảo trải nghiệm người dùng tốt.
* **Lợi ích:**
  + Giúp phát triển nhanh hơn, giảm lỗi về sau.
  + Dễ dàng mở rộng hoặc bảo trì hệ thống.
  + Lợi ích tăng lên nhưng vẫn chưa tạo ra giá trị kinh doanh trực tiếp.

**Giai đoạn 3: Phát triển giao diện & Tính năng (Chi phí: 18 triệu VND, Lợi ích: 50%)**

* **Công việc chính:**
  + Xây dựng giao diện chính của website, đảm bảo tính thân thiện và dễ sử dụng.
  + Phát triển các tính năng cốt lõi như đăng ký, đăng nhập, trang sản phẩm, giỏ hàng.
  + Tích hợp các dịch vụ thanh toán, quản lý đơn hàng, tài khoản khách hàng.
* **Lợi ích:**
  + Có sản phẩm thực tế để trình bày với người dùng và nhà đầu tư.
  + Bắt đầu có thể chạy thử nghiệm nội bộ.
  + Lợi ích tăng mạnh vì website đã có thể hoạt động ở mức cơ bản.

**Giai đoạn 4: Kiểm thử (Chi phí: 8 triệu VND, Lợi ích: 60%)**

* **Công việc chính:**
  + Kiểm thử từng chức năng để đảm bảo hoạt động ổn định.
  + Phát hiện và sửa lỗi hiệu suất, bảo mật.
  + Thử nghiệm với một nhóm người dùng giới hạn.
* **Lợi ích:**
  + Giúp giảm thiểu rủi ro khi triển khai chính thức.
  + Cải thiện hiệu suất và bảo mật, nâng cao trải nghiệm người dùng.
  + Lợi ích tiếp tục tăng nhưng chưa thể kiếm tiền từ người dùng chính thức.

**Giai đoạn 5: Triển khai & Bảo trì (Chi phí: 5 triệu VND, Lợi ích: 80%)**

* **Công việc chính:**
  + Đưa website vào hoạt động chính thức.
  + Thu thập phản hồi từ người dùng và cải thiện sản phẩm.
  + Xử lý lỗi phát sinh, tối ưu hóa hiệu suất.
  + Xây dựng chiến lược marketing để thu hút khách hàng.
* **Lợi ích:**
  + Bắt đầu tạo ra doanh thu từ người dùng thông qua quảng cáo, bán hàng hoặc dịch vụ đăng ký.
  + Website ổn định và hoạt động tốt, tạo dựng uy tín.
  + Lợi ích đạt mức cao nhất vì đã có lượng khách hàng sử dụng.

**Kết luận về tính khả thi kinh tế:**

* Tính khả thi kinh tế cao về lâu dài, mặc dù giai đoạn đầu có rủi ro và cần nguồn đầu tư lớn.
* Dự án có lợi nhuận bền vững sau khi vượt qua giai đoạn thử nghiệm và triển khai thực tế.
* Thời gian hoàn vốn hợp lý, thường từ năm 4 - năm 6, sau đó lợi ích sẽ tăng mạnh.

### 2. Tính khả thi về kỹ thuật

Bảng 1. Bảng phân tích tính khả thi về kỹ thuật

|  |  |  |  |
| --- | --- | --- | --- |
| **Thành phần** | **Hạ tầng hiện tại** | **Khả năng hỗ trợ** | **Cần nâng cấp?** |
| AI & NLP | Có GPT models (API OpenAI) | Hỗ trợ cơ bản | Cần tối ưu prompt hoặc fine-tuning |
| Backend | Server nhỏ hoặc VPS | Có thể chạy nhưng tải lớn sẽ chậm | Cần nâng cấp lên cloud hoặc cluster |
| Cơ sở dữ liệu | MySQL | Hỗ trợ tốt | Không cần nâng cấp ngay |
| Triển khai | Server vật lý hoặc VPS | Hỗ trợ giới hạn | Cần Docker/K8s để mở rộng |
| Lịch đăng bài | Cron jobs đơn giản | Hoạt động tốt | Có thể tích hợp Celery để linh hoạt hơn |
| Chia sẻ MXH | API có sẵn nhưng hạn chế | Hoạt động được | Cần tối ưu API tránh bị rate-limit |

**Kết luận về tính khả thi kỹ thuật:** Hệ thống AutoBlog AI có thể triển khai với hạ tầng hiện tại, nhưng để mở rộng và tối ưu chi phí, cần nâng cấp lên cloud, tối ưu AI và áp dụng các công nghệ quản lý backend tiên tiến.

## III. Tài liệu yêu cầu

### 1. Yêu cầu chức năng

Bảng 2. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Mô tả yêu cầu** | **Mức độ ưu tiên** | **Tiêu chí chấp nhận** |
| Quản lý tài khoản người dùng | Hệ thống cho phép đăng ký, đăng nhập, quên mật khẩu | Cao | Người dùng có thể tạo tài khoản, đăng nhập và đặt lại mật khẩu khi cần |
| Quản lý blog | Tạo, chỉnh sửa, xóa bài viết | Cao | Người dùng có thể tạo bài viết với tiêu đề, nội dung, hình ảnh |
| Lập lịch đăng bài | Hỗ trợ đặt lịch đăng bài tự động theo thời gian thiết lập | Cao | Người dùng có thể đặt lịch đăng bài và bài viết được tự động đăng đúng giờ |
| Hỗ trợ nhiều nền tảng | Đăng bài lên nhiều nền tảng như WordPress, Blogger, Medium | Cao | Hệ thống có thể đăng bài lên ít nhất 3 nền tảng khác nhau |
| Quản lý danh mục | Phân loại bài viết theo danh mục | Trung bình | Người dùng có thể tạo, chỉnh sửa và xóa danh mục |
| Tích hợp AI viết bài | Gợi ý tiêu đề, tạo nội dung bằng AI | Trung bình | Hệ thống cung cấp nội dung gợi ý phù hợp với chủ đề |
| Hệ thống báo cáo & thống kê | Thống kê lượt xem, tương tác với bài viết | Thấp | Người dùng có thể xem báo cáo tổng hợp về bài viết |

### 2. Yêu cầu phi chức năng

Bảng 3. Yêu cầu phi chức năng

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Mô tả yêu cầu** | **Mức độ ưu tiên** | **Tiêu chí chấp nhận** |
| Hiệu suất | Hệ thống có thể xử lý ít nhất 1000 bài viết/ngày | Cao | Hệ thống không bị gián đoạn khi xử lý nhiều bài viết |
| Bảo mật | Hỗ trợ xác thực hai lớp, mã hóa dữ liệu | Cao | Người dùng có thể bật 2FA và dữ liệu được bảo vệ |
| Tính mở rộng | Hệ thống có thể tích hợp thêm nền tảng blog mới | Trung bình | Có API mở để thêm nền tảng đăng bài mới |
| Giao diện người dùng | Giao diện trực quan, dễ sử dụng | Cao | Người dùng có thể thao tác mà không cần hướng dẫn |
| Khả năng tương thích | Hỗ trợ trên nhiều thiết bị | Trung bình | Hệ thống hoạt động tốt trên cả trình duyệt web và mobile |

## IV. Danh sách rủi ro

Bảng 4. Bảng danh sách rủi ro

|  |  |  |  |
| --- | --- | --- | --- |
| **Rủi ro tiềm ẩn** | | **Chiến lược phòng ngừa** | **Chiến lược ứng phó** |
| **Rủi ro kỹ thuật** | 1. Chất lượng nội dung AI tạo ra không đạt yêu cầu | - Lựa chọn mô hình AI có chất lượng và đã được kiểm chứng.  - Xác định rõ ràng dữ liệu đầu vào và chủ đề để AI tạo nội dung.  - Dành thời gian kiểm thử và tinh chỉnh mô hình AI. | - Chuẩn bị quy trình chỉnh sửa nội dung thủ công hiệu quả.  - Có phương án sửa dụng nội dung do người dùng tạo nếu chất lượng AI không đảm bảo. |
| 2. Tích hợp API AI gặp khó khăn hoặc hiệu suất kém | - Nghiên cứu kỹ API AI trước khi tích hợp.  - Lập kế hoạch dự phòng cho việc tích hợp API (ví dụ: sử dụng API thay thế hoặc tự phát triển mô-đun tạo nội dung cơ bản).  - Kiểm thử hiệu suất API trong môi trường dự án. | - Giảm bớt phạm vi chức năng AI nếu tích hợp API gặp nhiều vấn đề.  - Tập trung vào các chức năng cốt lõi khác của hệ thống. |
| 3. Lỗ hổng bảo mật (SQL Injection, XSS) | - Tuân thủ các nguyên tắc bảo mật trong quá trình phát triển (ví dụ: input validation, output encoding).  - Thực hiện kiểm thử bảo mật thường xuyên trong quá trình phát triển và sau triển khai. | - Xây dựng kế hoạch ứng phó sự cố bảo mật.  - Chuẩn bị sẵn sàng các bản vá lỗi bảo mật và quy trình cập nhật nhanh chóng. |
|  | 4. Hiệu suất hệ thống kém khi lượng người dùng lớn | - Tối ưu hóa mã nguồn và cấu trúc cơ sở dữ liệu từ giai đoạn thiết kế.  - Sử dụng các công nghệ và kỹ thuật tối ưu hiệu suất (ví dụ: caching, load balancing).  - Thực hiện kiểm thử hiệu suất và khả năng chịu tải. | - Chuẩn bị phương án mở rộng hệ thống (scaling) nếu cần thiết.  - Giám sát hiệu suất hệ thống liên tục sau triển khai và có biện pháp can thiệp kịp thời. |
|  | 5. Phụ thuộc vào công nghệ và phiên bản cụ thể (ReactJS, Django) | - Lựa chọn các công nghệ phổ biến và có cộng đồng hỗ trợ lớn.  - Theo dõi các bản cập nhật và thay đổi của công nghệ để có kế hoạch thích ứng. | - Chuẩn bị phương án chuyển đổi công nghệ nếu cần thiết trong tương lai (ví dụ: khi công nghệ hiện tại trở nên lỗi thời hoặc không được hỗ trợ). |
| **Rủi ro nguồn lực** | 6. Thiếu nhân sự có kỹ năng phù hợp | - Tuyển dụng sớm và có kế hoạch đào tạo nhân sự.  - Phân công công việc hợp lý dựa trên kỹ năng của từng thành viên.  - Sử dụng nguồn lực bên ngoài (freelancer, thuê chuyên gia tư vấn) nếu cần thiết. | - Điều chỉnh phạm vi dự án hoặc tiến độ nếu thiếu hụt nhân sự nghiêm trọng.  - Ưu tiên các chức năng cốt lõi để đảm bảo sản phẩm tối thiểu khả dụng . |
| 7. Khó khăn trong việc phối hợp và giao tiếp nhóm | - Thiết lập quy trình giao tiếp rõ ràng và hiệu quả (ví dụ: họp hàng ngày, sử dụng công cụ giao tiếp nhóm).  - Tổ chức các buổi team building để tăng cường tinh thần đồng đội. | - Có người quản lý dự án có kinh nghiệm để điều phối và giải quyết xung đột trong nhóm.  - Sử dụng công cụ quản lý dự án để theo dõi tiến độ và phân công công việc minh bạch. |
| 8. Trễ tiến độ trong các giai đoạn phát triển | - Lập kế hoạch chi tiết và thực tế cho từng giai đoạn.  - Theo dõi tiến độ dự án sát sao và có biện pháp can thiệp sớm khi có dấu hiệu chậm trễ.  - Sử dụng phương pháp phát triển linh hoạt (Agile) để dễ dàng thích ứng với thay đổi. | - Điều chỉnh tiến độ dự án (nếu có thể) hoặc giảm bớt phạm vi dự án để đảm bảo hoàn thành đúng thời hạn quan trọng nhất.  - Tăng cường nguồn lực (nếu có thể) hoặc làm thêm giờ để bù đắp thời gian bị chậm trễ. |
| **Rủi ro tài chính** | 9. Vượt quá ngân sách dự kiến (hosting, tên miền, AI API) | - Lập ngân sách chi tiết và có dự phòng cho các khoản chi phí phát sinh.  - Tìm kiếm các giải pháp hosting và tên miền với chi phí hợp lý.  - Tận dụng các gói miễn phí hoặc dùng thử của API AI. | - Cắt giảm các tính năng không cốt lõi để giảm chi phí phát triển.  - Tìm kiếm thêm nguồn tài trợ hoặc đầu tư nếu cần thiết. |
| 10. Chi phí phát sinh không lường trước | - Dự trù một khoản chi phí dự phòng (10-15% tổng ngân sách).   - Theo dõi chi tiêu thực tế và so sánh với ngân sách dự kiến thường xuyên. | - Ưu tiên các hạng mục chi tiêu quan trọng nhất và cắt giảm các hạng mục ít quan trọng hơn.   - Đàm phán lại với các nhà cung cấp để có giá ưu đãi hơn. |
| **Rủi ro phạm vi dự án** | 11. Thay đổi yêu cầu hoặc phát sinh yêu cầu mới (scope creep) | - Xác định yêu cầu rõ ràng và chi tiết ngay từ đầu giai đoạn "Xác định yêu cầu".  - Thiết lập quy trình quản lý thay đổi yêu cầu chặt chẽ.  - Đánh giá kỹ lưỡng tác động của mọi thay đổi yêu cầu đến tiến độ và ngân sách. | - Ưu tiên các yêu cầu cốt lõi và trì hoãn các yêu cầu bổ sung nếu chúng ảnh hưởng đến tiến độ dự án.  - Đàm phán với khách hàng về việc điều chỉnh tiến độ và ngân sách nếu có thay đổi yêu cầu lớn. |
| 12. Yêu cầu không rõ ràng trong giai đoạn "Xác định yêu cầu" | - Dành đủ thời gian và nguồn lực cho giai đoạn "Xác định yêu cầu".  - Sử dụng các kỹ thuật thu thập yêu cầu hiệu quả (ví dụ: phỏng vấn, khảo sát, workshop).  - Xác thực yêu cầu với khách hàng để đảm bảo hiểu đúng và đầy đủ. | - Linh hoạt trong giai đoạn "Phân tích" và "Thiết kế hệ thống" để điều chỉnh khi yêu cầu được làm rõ hơn.  - Dự trù thời gian cho việc làm rõ và điều chỉnh yêu cầu trong kế hoạch dự án. |
| **Rủi ro thời gian** | 13. Tiến độ 10 tuần quá ngắn cho tất cả chức năng | - Ưu tiên phát triển các chức năng cốt lõi trong phạm vi thời gian 10 tuần.  - Phân chia dự án thành các giai đoạn nhỏ hơn và có mục tiêu rõ ràng cho từng giai đoạn.  - Sử dụng các công cụ và phương pháp giúp tăng tốc độ phát triển (ví dụ: code generation, component libraries). | - Giảm bớt phạm vi dự án nếu nhận thấy tiến độ 10 tuần không khả thi.  - Đàm phán với khách hàng về việc kéo dài tiến độ dự án nếu cần thiết để đảm bảo chất lượng sản phẩm. |
| 14. Chậm trễ ở một giai đoạn ảnh hưởng đến toàn bộ tiến độ | - Quản lý tiến độ chặt chẽ ở từng giai đoạn.  - Xác định các giai đoạn quan trọng và có biện pháp phòng ngừa chậm trễ ở các giai đoạn này.  - Có kế hoạch dự phòng thời gian cho các giai đoạn có rủi ro cao. | - Ưu tiên hoàn thành các giai đoạn quan trọng trước.  - Tăng cường nguồn lực hoặc làm thêm giờ để bù đắp thời gian bị chậm trễ ở giai đoạn trước đó. |

## V. Kế hoạch ban đầu

### 1. Giai đoạn 1: Xác định yêu cầu (Tuần 1)

* Mục tiêu: Thu thập và đặc tả yêu cầu dự án.
* Công việc:
  + Thu thập yêu cầu từ khách hàng.
  + Xác định các tính năng chính cần phát triển.
* Sản phẩm: Tài liệu yêu cầu ban đầu.
* Nguồn lực:
  + *Nhân sự:* Quản lý dự án, chuyên viên phân tích thị trường, chuyên viên phân tích công nghệ
  + *Tài chính:* Chi phí họp nhóm, tài liệu khảo sát.(~ 2 triệu)
  + *Công nghệ:* Google Docs, Microsoft Teams để thảo luận, lưu trữ tài liệu trên Google Drive.

### 2. Giai đoạn 2: Phân tích (Tuần 2)

* Mục tiêu: Phân tích yêu cầu và lập tài liệu đặc tả chi tiết.
* Công việc:
  + Phân tích chi tiết yêu cầu hệ thống.
  + Lập tài liệu đặc tả yêu cầu (SRS).
  + Xác định các ràng buộc kỹ thuật.
* Sản phẩm: Tài liệu đặc tả yêu cầu (SRS).
* Nguồn lực:
  + *Nhân sự:* Quản lý dự án, chuyên viên phân tích thị trường, chuyên viên tư vấn công nghệ
  + *Tài chính:* Chi phí phân tích, công cụ quản lý dự án (~ 3 triệu).
  + *Công nghệ:* Jira, Confluence để quản lý yêu cầu, Notion để ghi chú.

### 3. Giai đoạn 3: Thiết kế hệ thống (Tuần 3)

* Mục tiêu: Xây dựng kiến trúc hệ thống và thiết kế giao diện người dùng.
* Công việc:
  + Xây dựng kiến trúc hệ thống.
  + Thiết kế giao diện người dùng (UI/UX).
  + Lên sơ đồ luồng dữ liệu và xử lý.
* Sản phẩm: Tài liệu thiết kế hệ thống, bản thiết kế UI/UX.
* Nguồn lực:
  + *Nhân sự:* kiến trúc sư hệ thống, chuyên viên thiết kế cơ sở dữ liệu, chuyên viên thiết kế giao diện người dùng
  + *Tài chính:* Chi phí phần mềm thiết kế, công cụ mô hình hóa (~ 7 triệu ).
  + *Công nghệ:* Figma, Draw.io để thiết kế giao diện và sơ đồ hệ thống.

### 4. Giai đoạn 4: Phát triển tính năng cốt lõi (Tuần 4 - 6)

* Mục tiêu: Xây dựng hệ thống tự động hóa đăng bài và quản lý nội dung.
* Công việc:
  + Xây dựng chức năng tự động hóa đăng bài.
  + Xây dựng chức năng quản lý nội dung (CRUD bài viết).
  + Xây dựng chức năng tìm kiếm nội dung.
  + Tạo API kết nối Front-end và Back-end.
* Sản phẩm: Các module chức năng cốt lõi (Back-end và Front-end), API.
* Nguồn lực:
  + *Nhân sự:* Frontend Developers, Backend Developer, AI Engineer, chuyên viên SEO
  + *Tài chính:* Chi phí thuê server, phần mềm phát triển (~18 triệu).
  + *Công nghệ:* React.js, Node.js, MongoDB, Firebase.

### 5. Giai đoạn 5: Tích hợp và kiểm thử (Tuần 7 - 8)

* Mục tiêu: Kiểm thử và tối ưu hóa hệ thống.
* Công việc:
  + Kiểm thử toàn bộ hệ thống (Unit Test, Integration Test, User Acceptance Test).
  + Sửa lỗi và tối ưu hiệu suất hệ thống.
* Sản phẩm: Hệ thống đã được kiểm thử và sửa lỗi.
* Nguồn lực:
  + *Nhân sự:* lập trình viên tích hợp, kỹ sư kiểm thử, chuyên gia bảo mật
  + *Tài chính:* Chi phí kiểm thử, công cụ test automation (~8 triệu).
  + *Công nghệ:* Selenium, Postman.

### 6. Giai đoạn 6: Triển khai (Tuần 9)

* Mục tiêu: Đưa hệ thống vào hoạt động thực tế.
* Công việc:
  + Đưa hệ thống lên server/hosting.
  + Cấu hình cơ sở dữ liệu, domain.
  + Kiểm tra hoạt động thực tế trên môi trường production.
* Sản phẩm: Hệ thống triển khai trên môi trường production.
* Nguồn lực:
  + *Nhân sự:* Kỹ sư Devops, kỹ sư hệ thống, Quản lý dự án.
  + *Tài chính:* Chi phí server, chứng chỉ SSL, domain (~5 triệu).
  + *Công nghệ:* AWS, Google Cloud, Docker.

### 7. Giai đoạn 7: Đánh giá và hoàn thiện (Tuần 10)

* Mục tiêu: Thu thập phản hồi và hoàn thiện hệ thống.
* Công việc:
  + Thu thập phản hồi từ người dùng thử nghiệm.
  + Sửa lỗi phát sinh và tối ưu hệ thống.
  + Cập nhật tài liệu hướng dẫn sử dụng.
* Sản phẩm: Hệ thống hoàn thiện, tài liệu hướng dẫn sử dụng.
* Nguồn lực:
  + *Nhân sự:* Chuyên viên phân tích dữ liệu, chuyên viên hỗ trợ khách hàng, lập trình viên bảo trì, quản lý dự án.
  + *Tài chính:* Chi phí khảo sát người dùng, bảo trì hệ thống (~3 triệu ).
  + *Công nghệ:* Google Forms (thu thập phản hồi), Notion (hướng dẫn sử dụng).

## VI. Đề xuất

### 1. Ước tính thời gian:

* Dự án dự kiến hoàn thành trong 10 tuần (từ 17/2/2025 đến 21/4/2025).
* Biểu đồ Gantt

A graph with blue squares

AI-generated content may be incorrect.

Hình 2. Biểu đồ Gantt

### 2. Dự toán chi phí

**2.1. Chi phí nhân sự *(ước tính theo ngày, tính trung bình)***

* Quản lý dự án (1 người) -> 375000VND/ 1 ngày/ 70 ngày
* Chuyên viên thiết kế giao diện người dùng (1 người) -> 250000 VND/ 1 ngày / 4 ngày
* Frontend Developer (2 người) -> 250000 VND/ 1 ngày / 7 ngày
* Backend Developer (1 người) -> 250000 VND/ 1 ngày / 7 ngày
* AI Engineer (1 người) -> 250000 VND/ 1 ngày /21 ngày
* Chuyên viên SEO (1 người) -> 250000 VND/ 1 ngày /7 ngày
* QA Tester (1 người) -> 250000 VND/ 1 ngày/ 14 ngày

⮕ Tổng chi phí nhân sự (ước tính 2,5 tháng): 43 000 000 VNĐ.

**2.2. Chi phí hạ tầng & công cụ**

* Tên miền: 300000 VNĐ/ năm.
* Hosting Frontend: Miễn phí (Vercel).
* Hosting Backend: Miễn phí (Render)
* Cơ sở dữ liệu (MySQL/PostgreSQL): Miễn phí
* Công cụ AI: Sử dụng API miễn phí
* ⮕ Tổng chi phí hạ tầng : 300000 VNĐ/ năm

**2.3. Chi phí dự phòng & công cụ khác**

* Chi phí dự phòng: 10 - 15% tổng ngân sách.
* Công cụ hỗ trợ: Figma (UI/UX), Jira (quản lý dự án), GitHub/GitLab (CI/CD), v.v.

### 3. Môi trường triển khai

* Frontend: ReactJS, triển khai trên Netlify/Vercel.
* Backend: Python (Django/FastAPI), triển khai trên Heroku/AWS/GCP.
* Cơ sở dữ liệu: MySQL/PostgreSQL (trên cloud hoặc server riêng).
* Yêu cầu phần mềm:
  + Python 3.8+, Django/Fast API.
  + ReactJS + các thư viện frontend.
  + Thư viện AI: TensorFlow, PyTorch, Hugging Face Transformers

# CHƯƠNG 2. TÀI LIỆU PHÂN TÍCH YÊU CẦU

## I. Đặc tả chức năng

### 1. Tổng quan về hệ thống

Dự án AutoBlog được khởi xướng nhằm giải quyết nhu cầu tự động hóa quy trình quản lý nội dung cho các blogger, nhà sáng tạo nội dung và doanh nghiệp. Mục tiêu chính là xây dựng một nền tảng web tích hợp trí tuệ nhân tạo (AI) để tạo, quản lý và đăng tải nội dung blog một cách hiệu quả. Báo cáo này sẽ trình bày chi tiết các yêu cầu chức năng và phi chức năng của hệ thống, cũng như kế hoạch dự án tổng thể.

### 2. Yêu cầu chức năng

* Tự động hóa đăng bài: Hệ thống sẽ sử dụng AI để tạo nội dung dựa trên từ khóa và chủ đề do người dùng cung cấp. Người dùng có thể chỉnh sửa và tối ưu hóa nội dung trước khi đăng tải. Hệ thống cũng sẽ hỗ trợ lên lịch đăng bài tự động và gửi thông báo khi bài viết được đăng thành công.
* Quản lý nội dung: Người dùng có thể thêm, chỉnh sửa, xóa và phân loại bài viết theo danh mục và thẻ tag. Hệ thống cũng sẽ hỗ trợ quản lý hình ảnh và các tệp đính kèm.
* Tìm kiếm và phân loại: Người dùng có thể tìm kiếm bài viết theo từ khóa, tiêu đề hoặc nội dung. Hệ thống cũng sẽ cung cấp các công cụ lọc và sắp xếp bài viết theo danh mục, thẻ tag hoặc ngày tháng.
* Yêu cầu phi chức năng: Hệ thống phải đảm bảo hiệu suất cao, bảo mật, khả năng sử dụng và độ tin cậy.

### 3. Trường hợp sử dụng

* Người viết blog sử dụng hệ thống để tạo và đăng bài tự động, theo dõi hiệu suất bài viết và tối ưu hóa nội dung.
* Quản trị web sử dụng hệ thống để quản lý nội dung và lịch đăng bài của nhiều người viết blog, phân tích báo cáo thống kê để đánh giá hiệu quả của chiến lược nội dung.
* Người đọc sử dụng hệ thống để xem, tìm kiếm và chia sẻ nội dung.

### 4. Tiêu chí chấp nhận

* Hệ thống hoạt động ổn định và không có lỗi nghiêm trọng.
* Nội dung tự động đáp ứng yêu cầu về chất lượng và SEO.
* Lịch đăng bài hoạt động chính xác.
* Chức năng tìm kiếm và phân loại hiệu quả.
* Giao diện thân thiện và bảo mật.

## II. Kế hoạch dự án cuối cùng

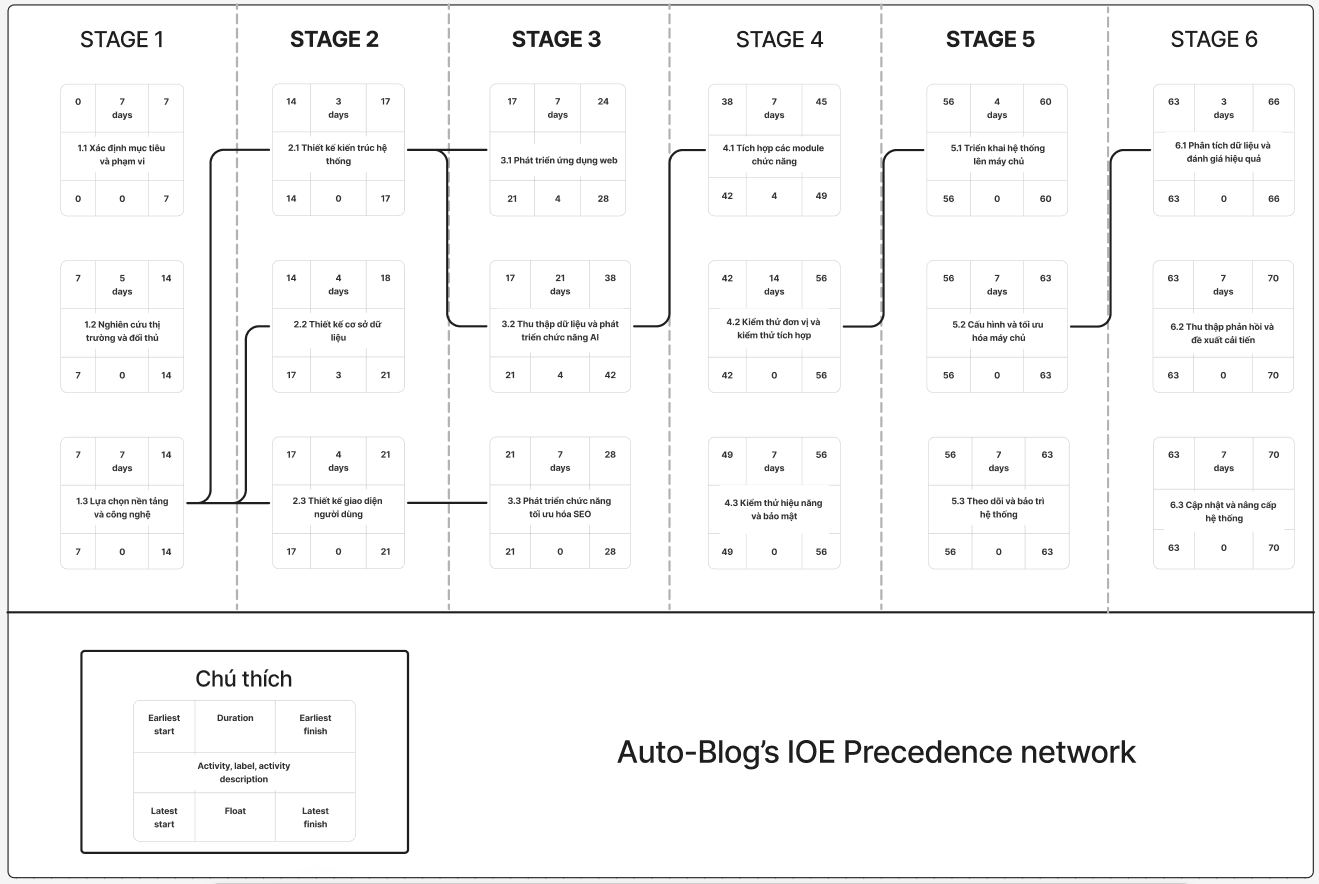
### 1. Phạm vi dự án

* Bao gồm:
  + Nền tảng web hoàn chỉnh:
    - Xây dựng một hệ thống web đa người dùng, có khả năng mở rộng để đáp ứng lượng truy cập tăng dần.
    - Phát triển giao diện người dùng (UI) trực quan và dễ sử dụng, tối ưu hóa cho trải nghiệm người dùng (UX) trên cả máy tính để bàn và thiết bị di động.
    - Triển khai hệ thống trên nền tảng đám mây để đảm bảo tính ổn định và khả năng mở rộng.
  + Tích hợp trí tuệ nhân tạo (AI):
    - Sử dụng các API của các mô hình AI có sẵn để tự động hóa việc tạo nội dung dựa trên từ khóa và chủ đề do người dùng cung cấp.
    - Phát triển các thuật toán để tối ưu hóa nội dung cho SEO, giúp tăng khả năng hiển thị trên các công cụ tìm kiếm.
    - Cung cấp các công cụ chỉnh sửa và tối ưu hóa nội dung, cho phép người dùng kiểm soát chất lượng và tính chính xác của nội dung được tạo tự động.
  + Hệ thống quản lý nội dung (CMS):
    - Phát triển các chức năng CRUD (Create, Read, Update, Delete) để quản lý bài viết, hình ảnh và các tệp đính kèm.
    - Cung cấp các công cụ phân loại nội dung theo danh mục và thẻ tag, giúp người dùng dễ dàng tổ chức và quản lý nội dung.
    - Xây dựng hệ thống quản lý người dùng, cho phép phân quyền và quản lý quyền truy cập của người dùng.
  + Công cụ tìm kiếm và phân loại:
    - Phát triển chức năng tìm kiếm toàn văn, cho phép người dùng tìm kiếm nội dung theo từ khóa, tiêu đề hoặc nội dung.
    - Cung cấp các công cụ lọc và sắp xếp nội dung theo danh mục, thẻ tag, ngày tháng hoặc các tiêu chí khác.
    - Xây dựng hệ thống gợi ý nội dung liên quan, giúp người dùng khám phá nội dung mới.
  + Bảo mật:
    - Triển khai các biện pháp bảo mật cơ bản để bảo vệ dữ liệu người dùng, bao gồm mã hóa dữ liệu, kiểm soát truy cập và phòng chống tấn công mạng.
    - Tuân thủ các tiêu chuẩn bảo mật phổ biến như OWASP.
* Không bao gồm:
  + Ứng dụng di động riêng biệt:
    - Dự án không bao gồm việc phát triển ứng dụng di động riêng biệt cho iOS hoặc Android.
    - Tuy nhiên, giao diện web sẽ được thiết kế để đáp ứng trên thiết bị di động.
  + Hỗ trợ đa ngôn ngữ:
    - Phiên bản đầu tiên của hệ thống chỉ hỗ trợ tiếng Anh.
    - Việc hỗ trợ đa ngôn ngữ sẽ được xem xét trong các phiên bản tiếp theo.
  + Tích hợp AI phức tạp:
    - Dự án không bao gồm việc tích hợp các công nghệ AI phức tạp như chatbot tự động trả lời hoặc phân tích cảm xúc người dùng.
    - Chỉ sử dụng các API có sẵn của các mô hình AI.
  + Marketing và quảng bá:
    - Dự án không bao gồm chiến lược marketing hoặc quảng bá trang blog đến người dùng cuối.
    - Đây là trách nhiệm của người dùng sau khi hệ thống được triển khai.
  + Thanh toán trực tuyến và bán hàng:
    - Hệ thống không tích hợp các tính năng thanh toán trực tuyến hoặc bán hàng qua trang blog.
  + Phát trực tiếp video/âm thanh:
    - Dự án không bao gồm tính năng phát trực tiếp video hoặc âm thanh.
  + Quản lý cộng đồng phức tạp:
    - Hệ thống không xây dựng các tính năng quản lý cộng đồng phức tạp như diễn đàn hoặc mạng xã hội.
  + Tùy chỉnh đặc thù:
    - Hệ thống sẽ cung cấp giải pháp tổng quát, không bao gồm các tùy chỉnh đặc thù cho từng khách hàng doanh nghiệp.

### 2. Các mốc quan trọng (Milestones) chi tiết

* Tuần 1: Giai đoạn Khởi động và Xác định Yêu cầu:
  + Kết quả: Hoàn thành thu thập yêu cầu từ các bên liên quan, xác định phạm vi dự án và phê duyệt tài liệu đề cương dự án.
* Tuần 2: Giai đoạn Phân tích Yêu cầu:
  + Kết quả: Hoàn thành tài liệu đặc tả yêu cầu chi tiết (SRS), bao gồm các trường hợp sử dụng, tiêu chí chấp nhận và yêu cầu phi chức năng, được phê duyệt bởi các bên liên quan.
* Tuần 3: Giai đoạn Thiết kế Hệ thống:
  + Kết quả: Hoàn thành thiết kế kiến trúc hệ thống tổng thể và thiết kế giao diện người dùng (UI/UX) chi tiết, được phê duyệt.
* Tuần 4-6: Giai đoạn Phát triển Tính năng Cốt lõi:
  + Kết quả: Hoàn thành phát triển các chức năng tự động hóa đăng bài, quản lý nội dung và tìm kiếm/phân loại.
* Tuần 7-8: Giai đoạn Tích hợp và Kiểm thử:
  + Kết quả: Hoàn thành tích hợp tất cả các thành phần hệ thống, thực hiện kiểm thử và sửa lỗi, hoàn thành kiểm thử chấp nhận người dùng (UAT).
* Tuần 9: Giai đoạn Triển khai:
  + Kết quả: Triển khai hệ thống lên môi trường sản xuất, cấu hình cần thiết và kiểm tra hoạt động thực tế.
* Tuần 10: Giai đoạn Đánh giá và Hoàn thiện:
  + Kết quả: Thu thập phản hồi người dùng, sửa lỗi, cập nhật tài liệu và bàn giao sản phẩm.

### 3. Dòng thời gian và phân bổ nguồn lực

**3.1. Sơ đồ thứ tự công việc** 

Hình 3. Sơ đồ thứ tự công việc

**3.2. Bảng danh sách yêu cầu nguồn lực**

(\*) Trong đó:

* F/T: fulltime



Hình 4. Danh sách yêu cầu nguồn lực

* Kỹ thuật:
  + Máy chủ đám mây: Render,Vercel.
  + Cơ sở dữ liệu: MySQL 8.0.
  + API AI: Hugging Face API.
  + Frontend: ReactJS 18, Redux Toolkit, Material UI 5.
  + Backend: Django 4.0, Python 3.9.
  + Công cụ quản lý dự án : Jira software, confluence, Slack.
* Tài chính:
  + Tổng ngân sách dự kiến: 50 Triệu VND (ước tính).
  + Ngân sách dự phòng: 7 Triệu VND (15% tổng ngân sách).
  + Các chi phí sẽ được theo dõi và báo cáo hàng tuần.
  + Các chi phí sẽ được phân bổ cụ thể cho từng hạng mục công việc.
  + Phân bổ chi phí theo giai đoạn: (Chi tiết đã được trình bày ở phần trước)

## III. Tài liệu đề xuất phát triển

### 1. Tóm tắt điều hành

AutoBlog là hệ thống tự động hóa quản lý và đăng bài blog, tích hợp AI để:

* Tạo nội dung từ khóa/chủ đề,
* Lên lịch đăng bài tự động,
* Quản lý dữ liệu đa nền tảng.

Dự án hướng đến đối tượng blogger, doanh nghiệp muốn giảm 50% thời gian quản lý nội dung thủ công, đồng thời đảm bảo tính bảo mật và trải nghiệm đa thiết bị.

### 2. Mục tiêu dự án

Bảng 5. Mục tiêu dự án

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mục tiêu | Chi tiết | Kết quả mong đợi |
| 1 | Tự động hóa đăng bài | Tạo nội dung bằng AI (GPT/Hugging Face), lên lịch và xuất bản tự động. | Giảm 50% thời gian quản lý thủ công. |
| 2 | Quản lý nội dung | Hỗ trợ CRUD (Thêm/Sửa/Xóa), phân loại theo tags/danh mục. | Dễ dàng tổ chức và tìm kiếm bài viết. |
| 3 | Tối ưu bảo mật | Mã hóa dữ liệu, phòng chống tấn công (SQL Injection, XSS). | Đạt tiêu chuẩn bảo mật cơ bản. |
| 4 | Giao diện đa nền tảng | Thiết kế responsive, tương thích mobile/desktop. | Trải nghiệm người dùng mượt mà. |

### 3. Phương pháp luận

**3.1. Cách tiếp cận phát triển hệ thống**

Hệ thống được phát triển theo phương pháp Agile, giúp đảm bảo tính linh hoạt, khả năng thích ứng với thay đổi và cải thiện hiệu suất làm việc nhóm. Quá trình phát triển sẽ chia thành nhiều sprint kéo dài 2 tuần, mỗi sprint tập trung vào một tập hợp các tính năng cụ thể và được kiểm thử liên tục để đảm bảo chất lượng.

**3.2. Quy trình phát triển chính:**

* Lập kế hoạch (Planning): Xác định yêu cầu, phân tích tính năng cần triển khai trong sprint.
* Phát triển (Development): Lập trình giao diện (Frontend), xử lý logic nghiệp vụ (Backend), tích hợp AI.
* Kiểm thử (Testing): Đánh giá tính năng, sửa lỗi và tối ưu hiệu suất.
* Triển khai (Deployment): Đưa sản phẩm lên môi trường thử nghiệm và sản phẩm.
* Phản hồi & Cải tiến (Feedback & Iteration): Điều chỉnh dựa trên phản hồi từ người dùng và thử nghiệm.

**3.3. Khuôn khổ và công nghệ sử dụng**

Bảng 6. Công nghệ sử dụng

|  |  |  |
| --- | --- | --- |
| Thành phần | Công nghệ | Mô tả |
| Frontend | React.js | Xây dựng giao diện thân thiện, responsive. |
| Backend | Django + MySQL | Xử lý logic nghiệp vụ, lưu trữ dữ liệu. |
| AI | Hugging Face | Xử lý NLP, tạo nội dung tự động. |
| Quản lý tiến độ | Jira | Theo dõi công việc, phân công nhiệm vụ. |
| Xử lý tác vụ nền | Celery | Lên lịch đăng bài, xử lý yêu cầu bất đồng bộ. |
| Hosting | Render (backend), Vercel (frontend) | Đảm bảo hệ thống hoạt động ổn định, có khả năng mở rộng. |

**3.4. Ước tính ngân sách**

Bảng 7. Ước tính ngân sách

|  |  |  |
| --- | --- | --- |
| Hạng mục | Chi phí dự kiến | Ghi chú |
| Tên miền | ~300 000 VNĐ/năm | Tùy nhà cung cấp (GoDaddy, Namecheap). |
| Hosting | Miễn phí | Render (backend), Vercel (frontend). |
| Công cụ AI | Miễn phí | Hugging Face API (nếu vượt giới hạn miễn phí). |
| Nhân công | 43 000 000 VNĐ | Chi phí trả cho nhân viên trong dự án |
| Dự phòng | 15% tổng ngân sách | Xử lý các chi phí phát sinh. |
| Tổng | ~50 000 000 VNĐ | Dựa trên các dự án tương tự. |

## IV. Thiết kế tổng thể

### 1. Thiết kế kiến trúc

A diagram of a service

AI-generated content may be incorrect.

Hình 5. Thiết kế kiến trúc

### 2. Sơ đồ luồng dữ liệu tổng thể

A diagram of a company

AI-generated content may be incorrect.

Hình 6. Sơ đồ luồng dữ liệu tổng thể

### 3. Điểm tích hợp

**3.1. Tích hợp Render (Triển khai backend)**

* Mục đích: Triển khai backend Django và xử lý tác vụ nền trên nền tảng Render .
* Phương thức:
  + Git Integration: Triển khai mã nguồn qua GitHub/GitLab.
  + Docker hoặc Native Deployment: Hỗ trợ cả container và môi trường sẵn có.
* Quy trình:
  + Cấu hình Render:
    - Đăng nhập vào Render Dashboard → Tạo Web Service mới.
    - Liên kết với repository chứa mã nguồn backend Django.
    - Định nghĩa biến môi trường:
      * SECRET\_KEY: Khóa bảo mật Django.
      * DATABASE\_URL: URL kết nối MySQL (tạo ở bước 2).
      * HUGGINGFACE\_API\_KEY: API key cho Hugging Face.
  + Triển khai mã nguồn:
    - Sử dụng render.yaml để định nghĩa build commands:

services:

  - type: web

    name: autoblog-backend

    env: python

    buildCommand: "pip install -r requirements.txt && python manage.py migrate"

    startCommand: "gunicorn autoblog.wsgi:application --bind 0.0.0.0:$PORT"

* Render tự động build và deploy khi có push lên nhánh chính.
* Tác vụ nền (Cron Job) :
  + Sử dụng Background Worker của Render để chạy Celery:

backgroundWorkers:

  - name: celery-worker

    command: "celery -A autoblog worker --loglevel=info"

**3.2. Tích hợp Render MySQL (Lưu trữ dữ liệu)**

* Mục đích: Sử dụng cơ sở dữ liệu MySQL được cung cấp bởi Render.
* Phương thức:
  + Tạo MySQL Database trên Render :
    - Vào Render Dashboard → New → MySQL Database .
    - Chọn phiên bản MySQL (ví dụ: 8.0) → Đặt tên database → Create Database
    - Sao chép External Database URL (ví dụ: mysql://user:[password@hostname.render.com](mailto:password@hostname.render.com):3306/dbname).
  + Cấu hình Django:

# settings.py

import dj\_database\_url

DATABASES = {

    'default': dj\_database\_url.config(

        default=os.getenv('DATABASE\_URL'),  # URL từ Render MySQL

        conn\_max\_age=600,

        conn\_health\_checks=True,

    )

}

* Cài đặt driver MySQL: Thêm mysqlclient (mysqlclient==2.2.0) vào requirements.txt

**Lưu ý khi dùng MySQL trên Render :**

* Kết nối an toàn:
  + Render tự động mã hóa kết nối qua SSL.
  + Cấu hình thêm SSL trong Django nếu cần:

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'dbname',

        'USER': 'user',

        'PASSWORD': os.getenv('DB\_PASSWORD'),

        'HOST': 'hostname.render.com',

        'PORT': '3306',

        'OPTIONS': {

            'ssl': {'ca': '/path/to/cert.pem'},  # Tùy chọn

        },

    }

}

* Backup dữ liệu: Render tự động backup MySQL hàng ngày → Khôi phục qua dashboard.
* Quản lý phiên bản: Render hỗ trợ MySQL 5.7, 8.0 → Chọn phiên bản tương thích với Django.

**3.3. Tích hợp Vercel (Triển khai frontend)**

* Mục đích: Deploy ứng dụng ReactJS lên Vercel.
* Phương thức:
  + Git Integration : Kết nối repository với Vercel.
  + CI/CD : Tự động build khi có commit.
* Cấu hình:

// vercel.json

{

  "version": 2,

  "builds": [

    { "src": "package.json", "use": "@vercel/static-build" }

  ],

  "routes": [{ "src": "/(.\*)", "dest": "/" }]

}

**3.4. Tích hợp Hugging Face API (Tạo nội dung tự động)**

* Mục đích: Gọi API Hugging Face để tạo nội dung từ backend Django.
* Phương thức: REST API (Sử dụng thư viện requests trong Django).
* Ví dụ :

import requests

def generate\_content(prompt):

    response = requests.post(

        "https://api.huggingface.co/models/gpt-3",

        headers={"Authorization": f"Bearer {os.getenv('HUGGINGFACE\_API\_KEY')}"},

        json={"input": prompt}

    )

    return response.json()["content"]

**3.5. Tích hợp Email Notifications**

* Mục đích: Gửi thông báo qua email khi bài viết được đăng.
* Phương thức: SMTP (Sử dụng SendGrid hoặc Gmail API).
* Cấu hình Django :

# settings.py

EMAIL\_BACKEND = "django.core.mail.backends.smtp.EmailBackend"

EMAIL\_HOST = "smtp.sendgrid.net"

EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

EMAIL\_HOST\_USER = "apikey"

EMAIL\_HOST\_PASSWORD = os.getenv("SENDGRID\_API\_KEY")

# CHƯƠNG 3. TÀI LIỆU ĐẶC TẢ THIẾT KẾ

## I. GIỚI THIỆU

### 1. Tổng quan

AutoBlog là hệ thống tự động hóa quản lý và đăng bài blog, tích hợp AI để:

* Tạo nội dung từ khóa/chủ đề,
* Lên lịch đăng bài tự động,
* Quản lý dữ liệu đa nền tảng.

Dự án hướng đến đối tượng blogger, doanh nghiệp muốn giảm 50% thời gian quản lý nội dung thủ công, đồng thời đảm bảo tính bảo mật và trải nghiệm đa thiết bị.

### 2. Phạm vi

* Bao gồm:
  + Nền tảng web hoàn chỉnh:
    - Xây dựng một hệ thống web đa người dùng, có khả năng mở rộng để đáp ứng lượng truy cập tăng dần.
    - Phát triển giao diện người dùng (UI) trực quan và dễ sử dụng, tối ưu hóa cho trải nghiệm người dùng (UX) trên cả máy tính để bàn và thiết bị di động.
    - Triển khai hệ thống trên nền tảng đám mây để đảm bảo tính ổn định và khả năng mở rộng.
  + Tích hợp trí tuệ nhân tạo (AI):
    - Sử dụng các API của các mô hình AI có sẵn để tự động hóa việc tạo nội dung dựa trên từ khóa và chủ đề do người dùng cung cấp.
    - Phát triển các thuật toán để tối ưu hóa nội dung cho SEO, giúp tăng khả năng hiển thị trên các công cụ tìm kiếm.
    - Cung cấp các công cụ chỉnh sửa và tối ưu hóa nội dung, cho phép người dùng kiểm soát chất lượng và tính chính xác của nội dung được tạo tự động.
  + Hệ thống quản lý nội dung (CMS):
    - Phát triển các chức năng CRUD (Create, Read, Update, Delete) để quản lý bài viết, hình ảnh và các tệp đính kèm.
    - Cung cấp các công cụ phân loại nội dung theo danh mục và thẻ tag, giúp người dùng dễ dàng tổ chức và quản lý nội dung.
    - Xây dựng hệ thống quản lý người dùng, cho phép phân quyền và quản lý quyền truy cập của người dùng.
  + Công cụ tìm kiếm và phân loại:
    - Phát triển chức năng tìm kiếm toàn văn, cho phép người dùng tìm kiếm nội dung theo từ khóa, tiêu đề hoặc nội dung.
    - Cung cấp các công cụ lọc và sắp xếp nội dung theo danh mục, thẻ tag, ngày tháng hoặc các tiêu chí khác.
    - Xây dựng hệ thống gợi ý nội dung liên quan, giúp người dùng khám phá nội dung mới.
  + Bảo mật:
    - Triển khai các biện pháp bảo mật cơ bản để bảo vệ dữ liệu người dùng, bao gồm mã hóa dữ liệu, kiểm soát truy cập và phòng chống tấn công mạng.
    - Tuân thủ các tiêu chuẩn bảo mật phổ biến như OWASP.
* Không bao gồm:
  + Ứng dụng di động riêng biệt:
    - Dự án không bao gồm việc phát triển ứng dụng di động riêng biệt cho iOS hoặc Android.
    - Tuy nhiên, giao diện web sẽ được thiết kế để đáp ứng trên thiết bị di động.
  + Hỗ trợ đa ngôn ngữ:
    - Phiên bản đầu tiên của hệ thống chỉ hỗ trợ tiếng Anh.
    - Việc hỗ trợ đa ngôn ngữ sẽ được xem xét trong các phiên bản tiếp theo.
  + Tích hợp AI phức tạp:
    - Dự án không bao gồm việc tích hợp các công nghệ AI phức tạp như chatbot tự động trả lời hoặc phân tích cảm xúc người dùng.
    - Chỉ sử dụng các API có sẵn của các mô hình AI.
  + Marketing và quảng bá:
    - Dự án không bao gồm chiến lược marketing hoặc quảng bá trang blog đến người dùng cuối.
    - Đây là trách nhiệm của người dùng sau khi hệ thống được triển khai.
  + Thanh toán trực tuyến và bán hàng:
    - Hệ thống không tích hợp các tính năng thanh toán trực tuyến hoặc bán hàng qua trang blog.
  + Phát trực tiếp video/âm thanh:
    - Dự án không bao gồm tính năng phát trực tiếp video hoặc âm thanh.
  + Quản lý cộng đồng phức tạp:
    - Hệ thống không xây dựng các tính năng quản lý cộng đồng phức tạp như diễn đàn hoặc mạng xã hội.
  + Tùy chỉnh đặc thù:
    - Hệ thống sẽ cung cấp giải pháp tổng quát, không bao gồm các tùy chỉnh đặc thù cho từng khách hàng doanh nghiệp.

## II. YÊU CẦU

### 1. Yêu cầu chức năng

Bảng 8. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Mô tả yêu cầu** | **Mức độ ưu tiên** | **Tiêu chí chấp nhận** |
| Quản lý tài khoản người dùng | Hệ thống cho phép đăng ký, đăng nhập, quên mật khẩu | Cao | Người dùng có thể tạo tài khoản, đăng nhập và đặt lại mật khẩu khi cần |
| Quản lý blog | Tạo, chỉnh sửa, xóa bài viết | Cao | Người dùng có thể tạo bài viết với tiêu đề, nội dung, hình ảnh |
| Lập lịch đăng bài | Hỗ trợ đặt lịch đăng bài tự động theo thời gian thiết lập | Cao | Người dùng có thể đặt lịch đăng bài và bài viết được tự động đăng đúng giờ |
| Hỗ trợ nhiều nền tảng | Đăng bài lên nhiều nền tảng như WordPress, Blogger, Medium | Cao | Hệ thống có thể đăng bài lên ít nhất 3 nền tảng khác nhau |
| Quản lý danh mục | Phân loại bài viết theo danh mục | Trung bình | Người dùng có thể tạo, chỉnh sửa và xóa danh mục |
| Tích hợp AI viết bài | Gợi ý tiêu đề, tạo nội dung bằng AI | Trung bình | Hệ thống cung cấp nội dung gợi ý phù hợp với chủ đề |
| Hệ thống báo cáo & thống kê | Thống kê lượt xem, tương tác với bài viết | Thấp | Người dùng có thể xem báo cáo tổng hợp về bài viết |

### 2. Yêu cầu phi chức năng

Bảng 9. Yêu cầu phi chức năng

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Mô tả yêu cầu** | **Mức độ ưu tiên** | **Tiêu chí chấp nhận** |
| Hiệu suất | Hệ thống có thể xử lý ít nhất 1000 bài viết/ngày | Cao | Hệ thống không bị gián đoạn khi xử lý nhiều bài viết |
| Bảo mật | Hỗ trợ xác thực hai lớp, mã hóa dữ liệu | Cao | Người dùng có thể bật 2FA và dữ liệu được bảo vệ |
| Tính mở rộng | Hệ thống có thể tích hợp thêm nền tảng blog mới | Trung bình | Có API mở để thêm nền tảng đăng bài mới |
| Giao diện người dùng | Giao diện trực quan, dễ sử dụng | Cao | Người dùng có thể thao tác mà không cần hướng dẫn |
| Khả năng tương thích | Hỗ trợ trên nhiều thiết bị | Trung bình | Hệ thống hoạt động tốt trên cả trình duyệt web và mobile |

## III. CÁC YẾU TỐ THIẾT KẾ

### 1. Thiết kế cơ sở dữ liệu

A diagram of a computer program

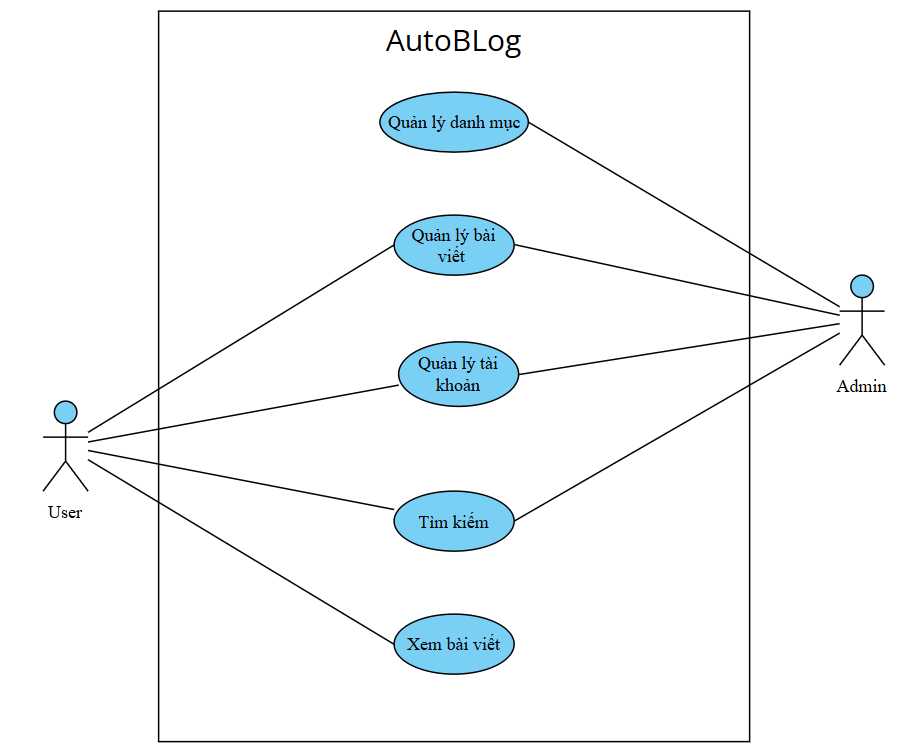
AI-generated content may be incorrect.

Hình 7. Sơ đồ ERD

### 2. Thiết kế hướng đối tượng

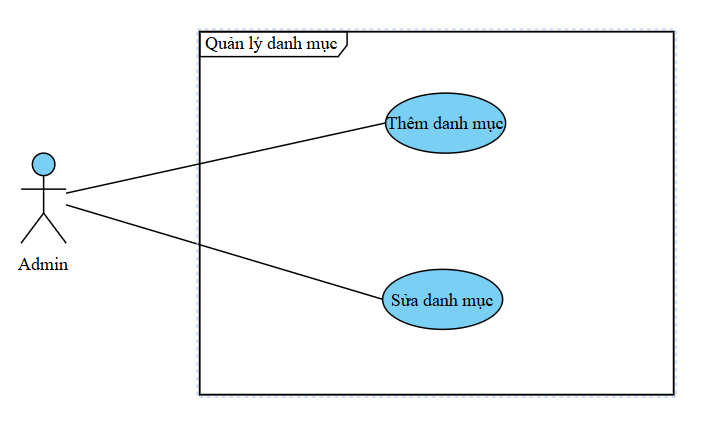
#### 2.1. Sơ đồ usecase

**2.1.1. Usecase tổng quan**



Hình 8. Sơ đồ usecase tổng quát

**2.1.2. Usecase Quản lý danh mục**



Hình 9. Sơ đồ usecase quản lý danh mục

Bảng 10. Đặc tả usecase quản lý danh mục

|  |  |
| --- | --- |
| **Tên usecase** | **Quản lý danh mục** |
| Mô tả | Admin có thể thực hiện các chức năng quản lý danh mục, bao gồm thêm danh mục mới và sửa danh mục đã có |
| Tác nhân chính | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào hệ thống. |
| Luồng sự kiện chính | 1. Admin truy cập vào chức năng Quản lý danh mục.  2. Hệ thống hiển thị danh sách danh mục hiện có.  3. Admin chọn một trong hai thao tác: Thêm danh mục hoặc Sửa danh mục.  4. Hệ thống thực hiện thao tác tương ứng:  - Nếu Admin chọn "Thêm danh mục":  + Hệ thống hiển thị form nhập thông tin danh mục mới.  + Admin nhập thông tin và xác nhận.  + Hệ thống kiểm tra dữ liệu hợp lệ và thêm danh mục vào hệ thống.  - Nếu Admin chọn "Sửa danh mục":  + Hệ thống hiển thị danh sách danh mục có thể sửa.  + Admin chọn danh mục cần sửa và điền thông tin  + Hệ thống xác nhận lại với Admin trước khi sửa.  + Nếu Admin đồng ý, hệ thống thực hiện sửa danh mục.  5. Hệ thống cập nhật danh sách danh mục và thông báo kết quả cho Admin. |
| Luồng phụ | - Thêm danh mục thành công: Hệ thống hiển thị danh mục mới trong danh sách.  - Sửa danh mục thành công: Hệ thống sửa danh mục và cập nhật giao diện. |
| Luồng ngoại lệ | - Lỗi nhập dữ liệu khi thêm danh mục: Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.  - Kết nối bị gián đoạn: Hệ thống hiển thị thông báo lỗi và yêu cầu thử lại sau. |
| Hậu điều kiện | - Nếu thêm danh mục thành công, danh mục mới xuất hiện trong danh sách.  - Nếu sửa danh mục thành công, danh mục sẽ được cập nhật.  - Nếu có lỗi, hệ thống hiển thị thông báo và không thay đổi dữ liệu. |

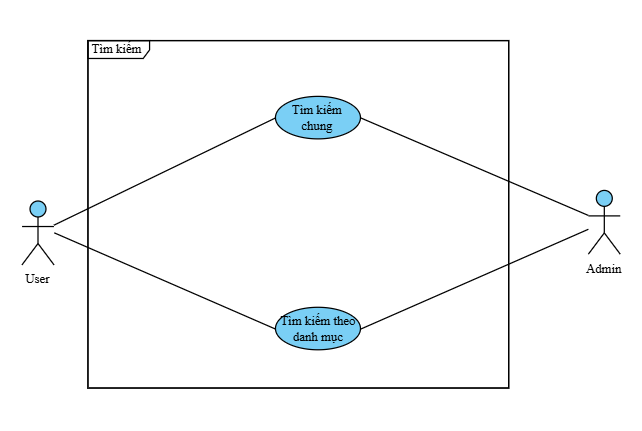
**2.1.3. Usecase Quản lý tài khoản**A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 10. Sơ đồ usecase quản lý tài khoản

Bảng 11. Đặc tả usecase quản lý tài khoản

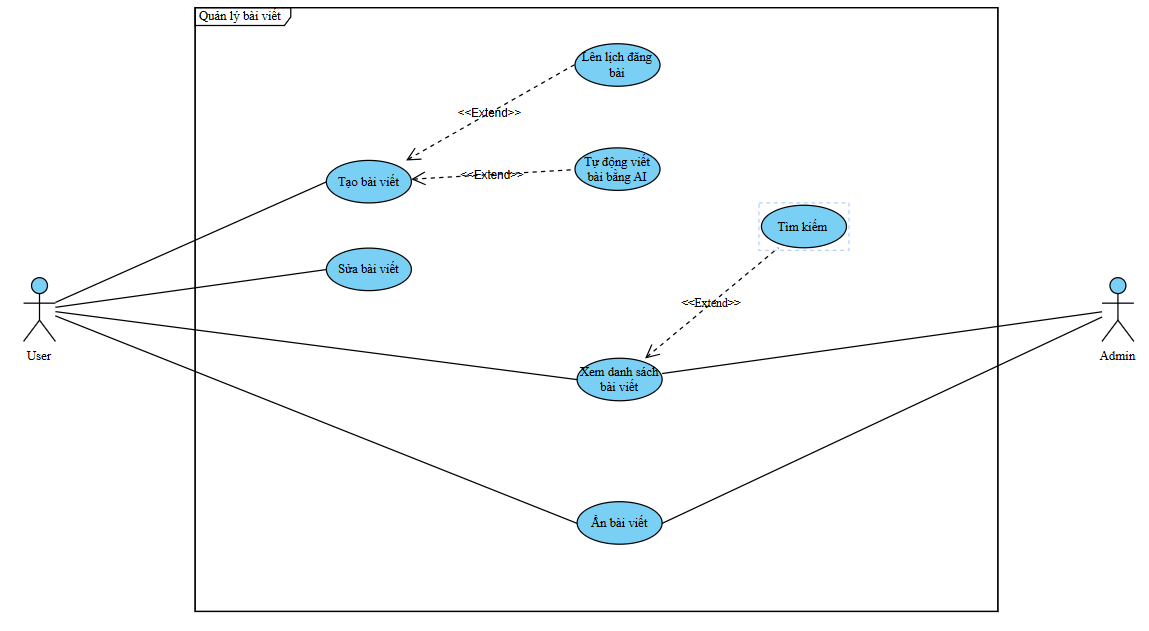
|  |  |
| --- | --- |
| **Tên usecase** | **Quản lý tài khoản** |
| Mô tả | Quản lý tài khoản cho phép người dùng và admin thực hiện các thao tác liên quan đến tài khoản như đăng nhập, đăng ký, quên mật khẩu, thay đổi thông tin, và quản lý tài khoản người dùng. |
| Tác nhân chính | User, Admin |
| Tiền điều kiện | User hoặc Admin có tài khoản hợp lệ hoặc có quyền tạo tài khoản. |
| Luồng sự kiện chính | 1. Người dùng hoặc admin truy cập vào hệ thống.  2. Người dùng có thể thực hiện các thao tác:  - Đăng nhập  - Đăng ký  - Quên mật khẩu  - Thay đổi thông tin cá nhân  - Thay đổi mật khẩu  3. Admin có thể thực hiện các thao tác:  - Thêm tài khoản  - Xem danh sách tài khoản  - Cấm tài khoản |
| Luồng phụ | - Thêm tài khoản thành công: Hệ thống hiển thị tài khoản mới trong danh sách.  - Thay đổi mật khẩu thành công: Hệ thống gửi thông báo về việc cập nhật mật khẩu.  - Cấm tài khoản thành công: Hệ thống khóa tài khoản và thông báo đến user. |
| Luồng ngoại lệ | Lỗi đăng nhập  - Người dùng nhập sai thông tin đăng nhập (sai username hoặc password).  - Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.  - Nếu nhập sai quá 5 lần, tài khoản bị khóa tạm thời.  Lỗi đăng ký  - Người dùng nhập username đã tồn tại.  - Hệ thống thông báo lỗi và yêu cầu chọn username khác.  - Nếu mật khẩu không đáp ứng tiêu chuẩn (quá ngắn, không có ký tự đặc biệt), hệ thống yêu cầu nhập lại.  Lỗi quên mật khẩu  - Người dùng nhập email không tồn tại trong hệ thống.  - Hệ thống thông báo lỗi và yêu cầu nhập lại email hợp lệ.  Lỗi thay đổi thông tin cá nhân  - Người dùng nhập thông tin không hợp lệ (ví dụ: email sai định dạng, số điện thoại không đúng).  - Hệ thống yêu cầu nhập lại thông tin hợp lệ.  Lỗi thêm tài khoản (Admin)  - Admin nhập username đã tồn tại.  - Hệ thống yêu cầu chọn username khác.  - Nếu nhập sai định dạng email, hệ thống thông báo lỗi. |
| Hậu điều kiện | Hệ thống cập nhật thông tin tài khoản thành công.  - Nếu có thay đổi, dữ liệu được lưu vào cơ sở dữ liệu. |

**2.1.4. Usecase Tìm kiếm**

Hình 11. Sơ dồ usecase tìm kiếm

Bảng 12. Đặc tả usecase tìm kiếm

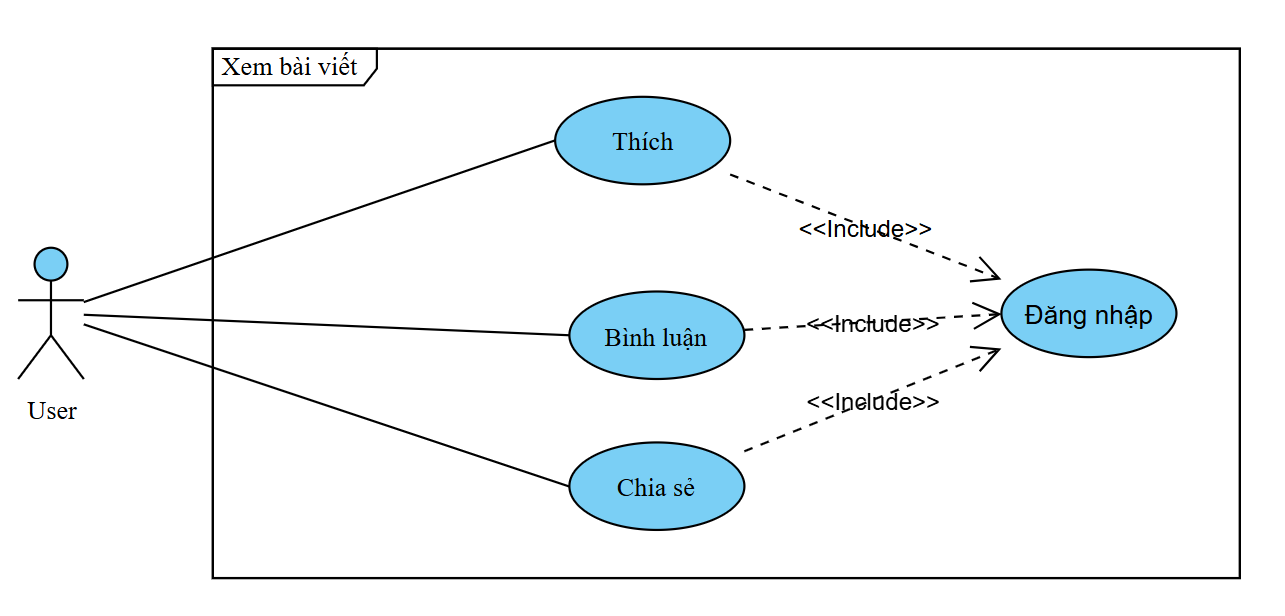
|  |  |
| --- | --- |
| **Tên usecase** | **Tìm kiếm** |
| Mô tả | Use Case này mô tả cách User và Admin thực hiện tìm kiếm thông tin trong hệ thống, bao gồm tìm kiếm chung theo từ khóa, tìm kiếm theo danh mục |
| Tác nhân chính | User, Admin |
| Tiền điều kiện | - Người dùng đã đăng nhập vào hệ thống. |
| Luồng sự kiện chính | Người dùng (User) hoặc Quản trị viên (Admin) truy cập vào chức năng tìm kiếm.  Hệ thống hiển thị giao diện tìm kiếm.  Người dùng hoặc Admin có thể thực hiện một trong hai thao tác: + Tìm kiếm chung: Nhập từ khóa vào ô tìm kiếm và thực hiện tìm kiếm. + Tìm kiếm theo danh mục: Chọn danh mục cụ thể, nhập từ khóa và thực hiện tìm kiếm. - Hệ thống xử lý yêu cầu, tìm kiếm dữ liệu trong cơ sở dữ liệu.  - Hệ thống hiển thị kết quả tìm kiếm phù hợp. |
| Luồng phụ | - Người dùng có thể sắp xếp hoặc lọc kết quả tìm kiếm. |
| Luồng ngoại lệ | - Không có kết quả tìm kiếm: Hệ thống hiển thị thông báo "Không tìm thấy kết quả phù hợp".  - Nhập từ khóa không hợp lệ: Hệ thống yêu cầu nhập lại thông tin tìm kiếm.  - Kết nối mạng bị lỗi: Hệ thống thông báo lỗi và yêu cầu thử lại. |
| Hậu điều kiện | - Nếu tìm kiếm thành công, hệ thống hiển thị kết quả phù hợp.  - Nếu không có kết quả, hệ thống thông báo cho người dùng. |

**2.1.5. Usecase quản lý bài viết** 

Hình 12. Sơ đồ usecase quản lý bài viết

Bảng 13. Đặc tả usecase quản lý bài viết

|  |  |
| --- | --- |
| **Tên usecase** | **Quản lý bài viết** |
| Mô tả | Use Case này mô tả quá trình người dùng (User) và quản trị viên (Admin) thực hiện các thao tác quản lý bài viết, bao gồm:  + Tạo bài viết (có thể lên lịch đăng hoặc sử dụng AI để viết).  + Chỉnh sửa bài viết.  + Xem danh sách bài viết (kèm theo tính năng tìm kiếm).  + Ẩn bài viết nếu cần thiết. |
| Tác nhân chính | User, Admin |
| Tiền điều kiện | - Người dùng đã có tài khoản và phải đăng nhập vào hệ thống trước khi thực hiện các thao tác liên quan đến bài viết. |
| Luồng sự kiện chính | - User hoặc Admin truy cập vào chức năng quản lý bài viết.  - Hệ thống hiển thị danh sách bài viết hiện có.  - Người dùng có thể thực hiện một trong các thao tác sau:  - Tạo bài viết:  + Nhập nội dung bài viết, tiêu đề, hình ảnh, danh mục.  + Chọn phương thức tạo bài viết (viết thủ công hoặc sử dụng AI).  + Có thể thiết lập lịch đăng bài.  + Lưu hoặc đăng bài viết.  - Sửa bài viết:  + Chọn bài viết cần chỉnh sửa.  + Thay đổi nội dung, tiêu đề, hình ảnh hoặc danh mục.  + Lưu lại bài viết đã chỉnh sửa.  - Xem danh sách bài viết:  + Xem danh sách tất cả bài viết đã tạo.  + Có thể tìm kiếm bài viết theo tiêu chí cụ thể.  - Ẩn bài viết:  + Chọn bài viết cần ẩn.  + Xác nhận thao tác.  + Bài viết sẽ không còn hiển thị cho người đọc nhưng vẫn tồn tại trong hệ thống. |
| Luồng phụ | **-** Luồng phụ 1: Lên lịch đăng bài  + Khi tạo bài viết, người dùng có thể chọn thời gian cụ thể để bài viết được tự động đăng.  - Luồng phụ 2: Sử dụng AI để viết bài  + Người dùng có thể nhập chủ đề và để hệ thống AI tự động tạo nội dung bài viết.  - Luồng phụ 3: Tìm kiếm bài viết  + Khi xem danh sách bài viết, người dùng có thể tìm kiếm bài viết bằng từ khóa hoặc bộ lọc theo danh mục. |
| Luồng ngoại lệ | - Nếu chưa đăng nhập, hệ thống yêu cầu người dùng đăng nhập trước khi thực hiện bất kỳ thao tác nào.  - Nếu có lỗi khi tạo bài viết (ví dụ: nội dung trống, lỗi kết nối AI), hệ thống hiển thị thông báo lỗi.  - Nếu bài viết không tồn tại hoặc không có quyền chỉnh sửa, hệ thống báo lỗi. |
| Hậu điều kiện | - Nếu tạo bài viết thành công, bài viết sẽ được hiển thị trong danh sách cá nhân.  - Nếu lên lịch đăng bài, bài viết sẽ được đăng theo thời gian đã chọn.  - Nếu sửa hoặc xóa bài viết thành công, hệ thống cập nhật thông tin bài viết. |

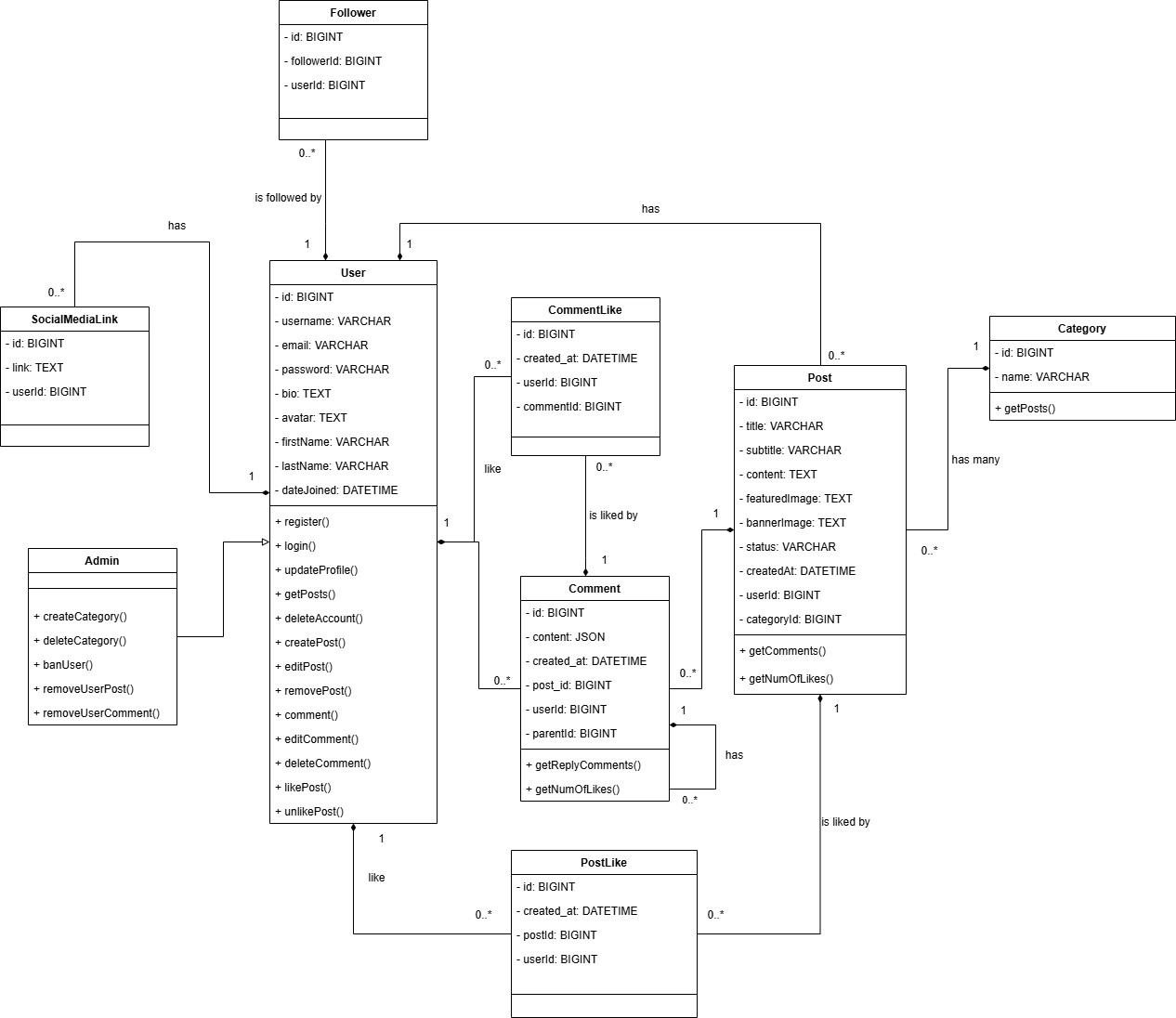
**2.1.6. Usecase Xem bài viết**

Hình 13. Sơ đồ usecase xem bài viết

Bảng 14. Đặc tả usecase xem bài viết

|  |  |
| --- | --- |
| **Tên usecase** | **Xem bài viết** |
| Mô tả | Hệ thống cho phép người dùng xem bài viết, đồng thời có thể thực hiện các thao tác mở rộng như thích, bình luận và chia sẻ bài viết. Một số thao tác yêu cầu người dùng phải đăng nhập. |
| Tác nhân chính | User |
| Tiền điều kiện | - Người dùng có thể truy cập hệ thống để xem bài viết mà không cần đăng nhập.  - Một số chức năng mở rộng như thích, bình luận và chia sẻ yêu cầu đăng nhập. |
| Luồng sự kiện chính | 1. Người dùng truy cập chức năng Xem bài viết.  2. Hệ thống hiển thị nội dung bài viết.  3. Người dùng có thể thực hiện các thao tác mở rộng như Thích, Bình luận, Chia sẻ. |
| Luồng phụ | - Khi người dùng muốn Thích bài viết, hệ thống kiểm tra trạng thái đăng nhập:    + Nếu chưa đăng nhập, hệ thống yêu cầu Đăng nhập.  - Khi người dùng muốn Bình luận, hệ thống kiểm tra trạng thái đăng nhập:    + Nếu chưa đăng nhập, hệ thống yêu cầu Đăng nhập  - Khi người dùng muốn Chia sẻ bài viết, hệ thống kiểm tra trạng thái đăng nhập:    + Nếu chưa đăng nhập, hệ thống yêu cầu Đăng nhập. |
| Luồng ngoại lệ | - Nếu bài viết không tồn tại hoặc đã bị xóa, hệ thống hiển thị thông báo lỗi.  - Nếu có lỗi khi thực hiện các thao tác (Thích, Bình luận, Chia sẻ), hệ thống hiển thị thông báo lỗi phù hợp. |
| Hậu điều kiện | - Nếu người dùng thích bài viết, số lượt thích của bài viết được cập nhật.  - Nếu người dùng bình luận, nội dung bình luận sẽ được lưu và hiển thị.  - Nếu người dùng chia sẻ bài viết, bài viết sẽ được chia sẻ lên nền tảng hoặc mạng xã hội phù hợp. |

#### 2.2. Sơ đồ lớp



Hình 14. Sơ đồ lớp

#### 2.3. Sơ đồ hoạt động

**2.3.1. Tạo bài viết**

**A black and white image of a map

AI-generated content may be incorrect.**

Hình 15. Sơ đồ hoạt động tạo bài viết

**2.3.2. Tìm kiếm**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 16. Sơ đồ hoạt động tìm kiếm

**2.3.3. Cập nhật bài viết**

**A diagram of a flowchart

AI-generated content may be incorrect.**

Hình 17. Sơ đồ hoạt động cập nhật bài viết

**2.3.4. Xoá bài viết**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 18. Sơ đồ hoạt động xoá bài viết

#### 2.4. Sơ đồ tuần tự

**2.4.1. Tạo bài viết**

A white sheet of paper with black text

AI-generated content may be incorrect.

Hình 19. Sơ đồ tuần tự tạo bài viết

**2. Tìm kiếm**

A diagram of a diagram

AI-generated content may be incorrect.

Hình 20. Sơ đồ tuần tự tìm kiếm

**2. Cập nhật bài viết**

A white sheet of paper with black text

AI-generated content may be incorrect.

Hình 21. Sơ đồ tuần tự cập nhật bài viết

**2. Xoá bài viết**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 22. Sơ đồ tuần tự xoá bài viết

### 3. Thiết kế UI/UX

Với dự án Auto Blog, mục tiêu của nhóm là tạo ra một nền tảng thân thiện với người dùng, nơi người dùng có thể dễ dàng tạo, quản lý và chia sẻ nội dung blog. Thiết kế UI/UX được xây dựng dựa trên sự hiểu biết sâu sắc về nhu cầu của người dùng, tập trung vào tính đơn giản, trực quan và hiệu quả.

#### 3.1. Wireframes:

* **Mục đích:**
  + Phác thảo cấu trúc cơ bản của giao diện người dùng, giúp xác định bố cục và luồng thông tin.
  + Tạo ra các bản phác thảo có độ trung thực thấp, tập trung vào chức năng và bố cục.
* **Phương pháp:**
  + Sử dụng Figma để tạo wireframes.
* **Các trang chính:**
  + **Trang Đăng nhập:**
    - Bố cục: Form đăng nhập đơn giản, tập trung vào email và mật khẩu.
    - Các yếu tố: Trường nhập email, mật khẩu, nút "Đăng nhập", liên kết "Quên mật khẩu", liên kết "Đăng ký".
  + **Trang Đăng ký:**
    - Bố cục: Form đăng ký với các trường thông tin cơ bản.
    - Các yếu tố: Trường nhập email, mật khẩu, xác nhận mật khẩu, tên, họ, nút "Đăng ký".
  + **Trang Xem Blog (New & Danh mục):**
    - Bố cục: Danh sách bài viết dạng thẻ, hiển thị tiêu đề, ảnh thu nhỏ và đoạn trích.
    - Các yếu tố: Danh sách bài viết, bộ lọc theo danh mục, tab "Mới nhất".
  + **Trang Chi tiết Bài Blog:**
    - Bố cục: Hiển thị nội dung bài viết chi tiết, cho phép tương tác.
    - Các yếu tố: Tiêu đề, nội dung, thông tin tác giả, ngày đăng, phần bình luận, nút "Thích", nút chia sẻ.
  + **Trang Cá nhân:**
    - Bố cục: Hiển thị thông tin người dùng và danh sách bài viết đã đăng.
    - Các yếu tố: Thông tin cá nhân, danh sách bài viết, nút chỉnh sửa thông tin.
  + **Trang Tạo Bài Viết:**
    - Bố cục: Trình soạn thảo văn bản trực quan, cho phép định dạng và tải ảnh.
    - Các yếu tố: Trường nhập tiêu đề, nội dung, ảnh đại diện, danh mục, nút "Đăng bài".
* **Link Wireframe:** [Wireframe design](https://www.figma.com/design/jxtgVCFmtRdq23YUj3Ijkt/S-Blog?node-id=3748-3663&t=wuAaWHnIWv4ohsFZ-1)

#### 3.2 Nguyên mẫu (Prototype):

* **Mục đích:**
  + Mô phỏng các tương tác và luồng người dùng, giúp đánh giá trải nghiệm người dùng.
  + Tạo ra mẫu thử tương tác, để mô phỏng cách người dùng sẽ tương tác với sản phẩm cuối cùng.
* **Phương pháp:**
  + Tạo nguyên mẫu tương tác trong Figma.
* **Các tương tác chính:**
  + Luồng đăng nhập và đăng ký.
  + Xem bài viết và tương tác .
  + Điều hướng giữa các trang.
* **Link Prototype:** [Prototype design](https://www.figma.com/proto/jxtgVCFmtRdq23YUj3Ijkt/S-Blog?node-id=3741-4059&p=f&t=jK2YFAsjsYX8IiCd-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=3741%3A4059&show-proto-sidebar=1)

#### 3.3. Các yếu tố UI/UX chính:

* **Tính đơn giản và trực quan:** Giao diện được thiết kế rõ ràng, dễ hiểu, giúp người dùng dễ dàng thực hiện các thao tác.
* **Tính nhất quán:** Các yếu tố UI được sử dụng nhất quán trên toàn bộ trang web, tạo cảm giác quen thuộc cho người dùng.
* **Tối ưu hóa trải nghiệm đọc:** Phông chữ, kích thước chữ và khoảng cách dòng được lựa chọn cẩn thận để đảm bảo khả năng đọc tốt.
* **Tương tác mượt mà:** Các tương tác được thiết kế để phản hồi nhanh chóng và mượt mà, tạo cảm giác thoải mái cho người dùng.
* **Tối ưu hóa đa thiết bị:** Giao diện được thiết kế linh hoạt, đảm bảo hiển thị tốt trên mọi thiết bị.

## IV. CÂN NHẮC TRIỂN KHAI

### 1. Ngăn xếp công nghệ

**1.1. Backend**

* Ngôn ngữ lập trình: Python
* Framework: Django (Django Rest Framework - DRF để xây dựng API)
* Cơ sở dữ liệu: MySQL (Render MySQL)
* Xử lý tác vụ nền: Celery với Redis
* Triển khai: Render
* Bảo mật: Mã hóa mật khẩu với BCrypt, sử dụng biến môi trường để quản lý thông tin nhạy cảm
* API bên ngoài: Hugging Face API (tạo nội dung tự động)

**1.2. Frontend**

* Ngôn ngữ lập trình: TypeScript
* Thư viện chính: React.js
* Triển khai: Vercel
* Quản lý trạng thái: React Context
* Giao tiếp với backend: Fetch API hoặc Axios

**1.3. DevOps & Triển khai**

* Hệ thống kiểm soát phiên bản: Git (GitHub/GitLab)
* CI/CD:
  + Backend: Render hỗ trợ Git Integration, tự động build khi có thay đổi
  + Frontend: Vercel tự động deploy sau mỗi commit
* Docker: Có thể sử dụng Docker cho backend nếu cần

**1.4. Công nghệ bổ trợ**

* Gửi email: SMTP với Gmail API
* Môi trường và biến môi trường: .env, Render Environment Variables
* Quản lý background jobs: Celery chạy trên Render Background Worker

### 2. Tích hợp

Hệ thống sẽ bao gồm nhiều thành phần tích hợp với nhau để hoạt động mượt mà:

**2.1. Tích hợp Backend và Cơ sở dữ liệu (Render Django + MySQL)**

* Backend Django sẽ kết nối với **Render MySQL** bằng **DATABASE\_URL**.
* Kết nối MySQL trên Render được bảo mật qua **SSL**.
* Cấu hình trong Django (settings.py):

import dj\_database\_url

DATABASES = {

'default': dj\_database\_url.config(

default=os.getenv('DATABASE\_URL'),

conn\_max\_age=600,

conn\_health\_checks=True,

)

}

* MySQL sẽ được tự động backup hàng ngày trên Render.

**Tích hợp Backend và Frontend (API Django + React.js)**

* React.js frontend sẽ giao tiếp với Django backend qua **REST API**.
* Sử dụng **fetch hoặc Axios** để gọi API từ frontend.
* URL API backend sẽ được lưu trong biến môi trường frontend (.env trên Vercel).

**Triển khai Backend trên Render**

* Render sẽ tự động build backend từ GitHub/GitLab repository.
* Sử dụng **Gunicorn** để chạy Django trên Render:

services:

- type: web

name: autoblog-backend

env: python

buildCommand: "pip install -r requirements.txt && python manage.py migrate"

startCommand: "gunicorn autoblog.wsgi:application --bind 0.0.0.0:$PORT"

* Sử dụng **Celery** để xử lý tác vụ nền trên Render Background Worker.

**Triển khai Frontend trên Vercel**

* Kết nối repository frontend với Vercel.
* Cấu hình tự động build frontend sau mỗi commit.
* Cấu hình Vercel để xử lý routing SPA:

{

"version": 2,

"builds": [

{ "src": "package.json", "use": "@vercel/static-build" }

],

"routes": [{ "src": "/(.\*)", "dest": "/" }]

}

**4.5. Tích hợp AI từ Hugging Face**

* Backend Django sẽ gọi API từ Hugging Face để tạo nội dung tự động.
* Sử dụng thư viện requests để gửi yêu cầu đến Hugging Face API:

import requests

import os

def generate\_content(prompt):

response = requests.post(

"https://api.huggingface.co/models/gpt-3",

headers={"Authorization": f"Bearer {os.getenv('HUGGINGFACE\_API\_KEY')}"},

json={"input": prompt}

)

return response.json()["content"]

**4.6. Tích hợp Email Notification**

* Khi có bài viết mới, hệ thống sẽ gửi email thông báo qua **GMAIL SMTP**.
* Cấu hình Django để gửi email:

EMAIL\_BACKEND = "django.core.mail.backends.smtp.EmailBackend"

EMAIL\_HOST = "smtp.gmail.com"

EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

EMAIL\_HOST\_USER = "your\_email@gmail.com" # Thay bằng email của bạn

EMAIL\_HOST\_PASSWORD = os.getenv("GMAIL\_APP\_PASSWORD") # Sử dụng mật khẩu ứng dụng

## V. KIỂM TRA VÀ XÁC NHẬN

### 1. Chiến lược thử nghiệm

Để đảm bảo hệ thống hoạt động chính xác và ổn định, chiến lược kiểm thử sẽ bao gồm các giai đoạn sau:

**1.1. Kiểm thử đơn vị (Unit Testing)**

* Mục tiêu: Xác minh từng thành phần nhỏ (hàm, module) hoạt động đúng như mong đợi.
* Công cụ: Sử dụng pytest hoặc unittest trong Django backend, và Jest cho frontend React.js.
* Phạm vi kiểm thử:
  + Kiểm tra đăng ký, đăng nhập, và đặt lại mật khẩu.
  + Kiểm tra API tạo, chỉnh sửa, xóa bài viết.
  + Kiểm tra chức năng đặt lịch đăng bài.
  + Kiểm tra AI gợi ý tiêu đề và nội dung bài viết.

**1.2. Kiểm thử tích hợp (Integration Testing)**

* Mục tiêu: Đảm bảo các thành phần hoạt động tốt khi kết hợp với nhau.
* Công cụ: Postman, pytest-django, Selenium.
* Phạm vi kiểm thử:
  + Kiểm tra luồng đăng nhập.
  + Kiểm tra tính tương tác giữa Django backend và MySQL database.
  + Kiểm tra việc gọi API của Hugging Face để tạo nội dung AI.
  + Kiểm tra tính năng đặt lịch đăng bài với Celery.
  + Kiểm tra tích hợp email thông báo khi bài viết được đăng.

**1.3. Kiểm thử giao diện người dùng (UI/UX Testing)**

* Mục tiêu: Đảm bảo giao diện thân thiện, dễ sử dụng.
* Công cụ: Selenium, Lighthouse (Google Chrome DevTools).
* Phạm vi kiểm thử:
  + Kiểm tra khả năng thao tác trên giao diện web (React.js).
  + Kiểm tra khả năng phản hồi nhanh trên các thiết bị khác nhau (desktop, mobile, tablet).
  + Kiểm tra tốc độ tải trang, tối ưu hiển thị hình ảnh và văn bản.

**1.4. Kiểm thử hiệu suất (Performance Testing)**

* Mục tiêu: Đảm bảo hệ thống có thể xử lý khối lượng lớn dữ liệu.
* Công cụ: Apache JMeter, Locust.
* Phạm vi kiểm thử:
  + Kiểm tra khả năng xử lý 1000 bài viết/ngày.
  + Kiểm tra mức sử dụng CPU, RAM khi nhiều người dùng truy cập đồng thời.
  + Kiểm tra thời gian phản hồi của API trong điều kiện tải cao.

**1.5. Kiểm thử bảo mật (Security Testing)**

* Mục tiêu: Bảo vệ dữ liệu và ngăn chặn tấn công mạng.
* Công cụ: OWASP ZAP, SonarQube.
* Phạm vi kiểm thử:
  + Kiểm tra mã hóa mật khẩu bằng BCrypt.
  + Kiểm tra khả năng bảo vệ chống tấn công SQL Injection và XSS.
  + Kiểm tra bảo mật API và xác thực JWT.

**1.6. Kiểm thử chấp nhận người dùng (User Acceptance Testing - UAT)**

* Mục tiêu: Đảm bảo hệ thống đáp ứng nhu cầu thực tế của người dùng.
* Công cụ: Khảo sát người dùng, thử nghiệm trực tiếp.
* Phạm vi kiểm thử:
  + Người dùng thử nghiệm đăng ký, đăng nhập, đặt lịch bài viết.
  + Người dùng thao tác trên giao diện để kiểm tra tính trực quan.
  + Đánh giá trải nghiệm khi đăng bài lên nhiều nền tảng.

### 2. Tiêu chí xác nhận

**2.1. Yêu cầu chức năng**

Bảng 15. Tiêu chí xác nhận yêu cầu chức năng

|  |  |
| --- | --- |
| **Chức năng** | **Tiêu chí xác nhận** |
| Quản lý tài khoản | Người dùng có thể đăng ký, đăng nhập và khôi phục mật khẩu thành công. |
| Quản lý blog | Người dùng có thể tạo, chỉnh sửa, xóa bài viết với tiêu đề, nội dung, hình ảnh. |
| Lập lịch đăng bài | Người dùng có thể đặt lịch và bài viết được đăng đúng thời gian. |
| Quản lý danh mục | Người dùng có thể tạo, chỉnh sửa và xóa danh mục bài viết. |
| Tích hợp AI viết bài | Hệ thống cung cấp gợi ý nội dung phù hợp với chủ đề nhập vào. |
| Hệ thống báo cáo & thống kê | Người dùng có thể xem thống kê về lượt xem và tương tác bài viết. |

**2.2. Yêu cầu phi chức năng**

Bảng 16. Tiêu chí xác nhận yêu cầu phi chức năng

|  |  |
| --- | --- |
| **Tiêu chí** | **Tiêu chí xác nhận** |
| Hiệu suất | Hệ thống xử lý ít nhất 1000 bài viết/ngày mà không bị gián đoạn. |
| Bảo mật | Hệ thống hỗ trợ mã hóa dữ liệu bằng BCrypt. |
| Tính mở rộng | Có API mở cho phép tích hợp thêm nền tảng đăng bài mới. |
| Giao diện người dùng | Người dùng có thể dễ dàng thao tác mà không cần hướng dẫn. |
| Khả năng tương thích | Hệ thống hoạt động tốt trên trình duyệt web và mobile. |

# CHƯƠNG 4. TÀI LIỆU THỰC HIỆN

## US01 – Đăng ký/Đăng nhập

### 1.1. Mô tả User Story

**Mã User Story:** US01

**Tên chức năng:** Đăng ký/Đăng nhập

**Mục đích:** Thực hiện chức năng đăng ký và đăng nhập để người dùng có thể truy cập và sử dụng các tính năng của AutoBlog.

**Câu chuyện người dùng:**

* Là người dùng, tôi muốn đăng ký và đăng nhập vào hệ thống để sử dụng các chức năng của AutoBlog.

**Acceptance Criteria:**

* + Đăng ký tài khoản mới thành công khi cung cấp đầy đủ thông tin hợp lệ.
  + Đăng nhập thành công khi nhập đúng email và mật khẩu đã đăng ký.
  + Quản lý phiên đăng nhập bằng token (access/refresh), đảm bảo bảo mật.
  + Hiển thị lỗi nếu đăng nhập sai thông tin hoặc tài khoản bị khóa

**Độ phức tạp (Story Point):** 18

**Độ ưu tiên:** Cao

**Có liên quan bảo mật:** Yes

### 1.2. Thiết kế chức năng theo User Story

**Wireframe/Mockup:**

* Trang đăng ký: Form gồm tên, email, mật khẩu, xác nhận mật khẩu.
* Trang đăng nhập: Form gồm email, mật khẩu.

**Giao diện đầu vào/đầu ra**

* Đầu vào (Form)
* name: string
* email: string
* password: string
* confirmPassword: string (chỉ trong đăng ký)
* Đầu ra
  + 200 OK với thông tin người dùng và access toke
  + 400 Bad Request nếu thông tin không hợp lệ
  + 401 Unauthorized nếu sai thông tin đăng nhập

**Ghi chú kỹ thuật:**

* Hash mật khẩu bằng bcrypt
* Sử dụng JWT cho quản lý phiên
* Áp dụng chuẩn mã hóa HTTPS
* Framework: Django(BE), Reactjs (FE)

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng tạo tài khoản và đăng nhập để sử dụng các dịch vụ của AutoBlog.

**Bước sử dụng**

* Truy cập màn hình đăng ký hoặc đăng nhập
* Nhập thông tin vào các trường bắt buộc
* Nhấn nút “Đăng ký” hoặc “Đăng nhập”
* Hệ thống phản hồi thành công hoặc lỗi

**Hình ảnh minh họa**

Screens screenshot of a computer

AI-generated content may be incorrect.

Hình 23. Đăng nhập

Screens screenshot of a black and white screen

AI-generated content may be incorrect.

Hình . Đăng ký

**Tooltip & hỗ trợ:**

* Tooltip cho trường mật khẩu: “Mật khẩu tối thiểu 8 ký tự, bao gồm chữ và số”
* Hiển thị lỗi cụ thể ngay dưới mỗi trường không hợp lệ

## US02 – Tạo bài viết mới

### 1.1. Mô tả User Story

**Mã User Story:** US02

**Tên chức năng:** Tạo bài viết mới

**Mục đích:** Cho phép người dùng tạo bài viết mới để chia sẻ nội dung trên blog cá nhân.

**Câu chuyện người dùng:***Là người dùng, tôi muốn tạo một bài viết mới để đăng lên blog của mình.*

**Acceptance Criteria:**

* Cho phép nhập tiêu đề, nội dung và chọn hình ảnh minh họa.
* Lưu bài viết thành công vào cơ sở dữ liệu
* Hiển thị thông báo tạo thành công.
* Hiển thị lỗi nếu thiếu thông tin bắt buộc hoặc lỗi hệ thống.
* Hình ảnh được upload thành công và liên kết với bài viết.
* Có thể xem lại bài viết ngay sau khi tạo.

**Độ phức tạp (Story Point):** 20

**Độ ưu tiên:** Cao

**Có liên quan bảo mật:** Yes (bài viết cần gắn với đúng người dùng, xác thực token)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Form tạo bài viết: Gồm các trường tiêu đề, nội dung, hình ảnh, nút lưu.

**Giao diện đầu vào/đầu ra:**

* **Đầu vào (Form):**
* title: string – Tiêu đề bài viết
* content: string – Nội dung bài viết (HTML hoặc Markdown)
* image: File – Hình ảnh đính kèm (tuỳ chọn)
* **Đầu ra:**
* 201 Created nếu lưu thành công
* 400 Bad Request nếu thiếu tiêu đề hoặc nội dung
* 500 Internal Server Error nếu có lỗi hệ thống

**Ghi chú kỹ thuật:**

* Lưu hình ảnh vào cloud storage (VD: AWS S3 / Firebase)
* Validate input phía client và server
* Mã hóa token và kiểm tra quyền trước khi tạo bài
* Framework sử dụng: Django (BE), ReactJS (FE)

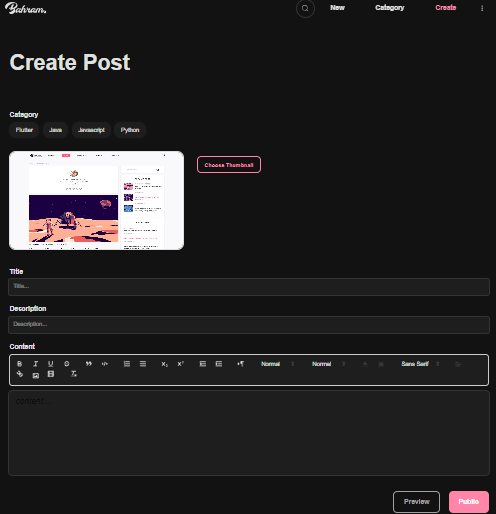
### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng cho phép người dùng tạo và chia sẻ nội dung mới trên blog cá nhân.

**Bước sử dụng:**

* Truy cập trang “Tạo bài viết”
* Nhập tiêu đề và nội dung
* (Tuỳ chọn) Chọn hình ảnh để minh họa
* Nhấn “Đăng bài”
* Xem bài viết mới trên trang cá nhân

**Hình ảnh minh họa:**



Hình 25. Tạo bài viết

**Tooltip & hỗ trợ:**

* Tooltip cho trường ảnh: “Chỉ hỗ trợ JPG, PNG, tối đa 5MB”
* Báo lỗi trực tiếp dưới các trường không hợp lệ

## US03 – Tự động tạo nội dung AI

### 1.1. Mô tả User Story

* **Mã User Story:** US03
* **Tên chức năng:** Tự động tạo nội dung AI
* **Mục đích:** Cung cấp khả năng tạo nhanh nội dung blog thông qua trí tuệ nhân tạo, giúp tiết kiệm thời gian viết bài và nâng cao trải nghiệm người dùng.
* **Câu chuyện người dùng:** *Là người dùng, tôi muốn hệ thống tự động tạo nội dung dựa trên từ khóa và chủ đề tôi cung cấp.*
* **Acceptance Criteria:**
  + Cho phép người dùng nhập từ khóa hoặc chủ đề.
  + AI sinh ra nội dung bài viết phù hợp với yêu cầu.
  + Người dùng có thể xem và chỉnh sửa nội dung được tạo.
  + Có thể lưu nội dung AI thành bài viết như bình thường.
  + Hiển thị lỗi nếu nhập thiếu hoặc yêu cầu không hợp lệ.
  + Thời gian phản hồi hợp lý (≤ 10 giây).
* **Độ phức tạp (Story Point):** 25
* **Độ ưu tiên:** Cao
* **Có liên quan bảo mật:** Yes (sử dụng AI qua API key, cần bảo mật và kiểm soát)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Giao diện nhập từ khóa/chủ đề + nút "Tạo nội dung"
* Vùng xem nội dung được tạo
* Nút "Lưu bài viết" hoặc "Chỉnh sửa trước khi lưu"

**Giao diện đầu vào/đầu ra:**

**Đầu vào (Form):**

* topic: string – Chủ đề tổng thể
* keywords: string[] – Danh sách từ khóa gợi ý

**Đầu ra:**

* 200 OK với nội dung được sinh tự động
* 400 Bad Request nếu thiếu dữ liệu đầu vào
* 500 Internal Server Error nếu AI service bị lỗi

**Ghi chú kỹ thuật:**

* Giao tiếp với AI thông qua OpenAI API hoặc LLM nội bộ
* Gọi API ở chế độ async, loading indicator khi đang xử lý
* Đảm bảo không sinh nội dung có yếu tố độc hại, vi phạm
* Ngôn ngữ hỗ trợ: Tiếng Việt và Tiếng Anh

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Tính năng giúp người dùng tạo nội dung blog nhanh chóng bằng trí tuệ nhân tạo dựa trên chủ đề và từ khóa gợi ý.

**Bước sử dụng:**

1. Truy cập mục “Tạo nội dung AI”
2. Nhập chủ đề và từ khóa
3. Nhấn “Tạo nội dung”
4. Xem nội dung sinh tự động
5. (Tuỳ chọn) Chỉnh sửa nội dung
6. Lưu bài viết

**Hình ảnh minh họa:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 26. AI tạo nội dung bài viết

**Tooltip & hỗ trợ:**

* Tooltip cho ô từ khóa: “Nhập từ khóa ngăn cách bằng dấu phẩy”
* Gợi ý mẫu chủ đề thông dụng bên dưới ô nhập

## US04 – Chỉnh sửa bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US04

**Tên chức năng:** Chỉnh sửa bài viết

**Mục đích:** Cho phép người dùng cập nhật nội dung bài viết đã đăng để sửa lỗi hoặc bổ sung thông tin mới.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn chỉnh sửa nội dung bài viết để cập nhật hoặc sửa lỗi khi cần thiết.*

**Acceptance Criteria:**

* Hiển thị form với thông tin bài viết hiện tại (tiêu đề, nội dung, hình ảnh).
* Cho phép chỉnh sửa các trường đã nhập.
* Lưu thay đổi thành công → thay thế nội dung cũ.
* Hiển thị thông báo khi cập nhật thành công hoặc lỗi.
* Kiểm tra quyền chỉnh sửa (chỉ chủ sở hữu bài viết mới được sửa).
* Ghi nhận thời gian cập nhật cuối cùng (lastUpdated).

**Độ phức tạp (Story Point):** 15  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (kiểm tra quyền người dùng và xác thực JWT)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Form chỉnh sửa bài viết tương tự form tạo bài viết
* Các trường đã có dữ liệu sẵn từ bài viết gốc
* Nút “Lưu thay đổi” và “Hủy bỏ”

**Giao diện đầu vào/đầu ra:**

* **Đầu vào (Form):**
* title: string – Tiêu đề cập nhật
* content: string – Nội dung mới
* image: File – Hình ảnh (có thể thay đổi hoặc giữ nguyên)
* **Đầu ra:**
* 200 OK nếu cập nhật thành công
* 400 Bad Request nếu thiếu tiêu đề/nội dung
* 403 Forbidden nếu người dùng không có quyền sửa
* 404 Not Found nếu bài viết không tồn tại

**Ghi chú kỹ thuật:**

* Sử dụng phương thức PUT hoặc PATCH
* Kiểm tra quyền sở hữu trước khi cho phép cập nhật
* Nếu thay hình ảnh, xóa hình cũ trên storage

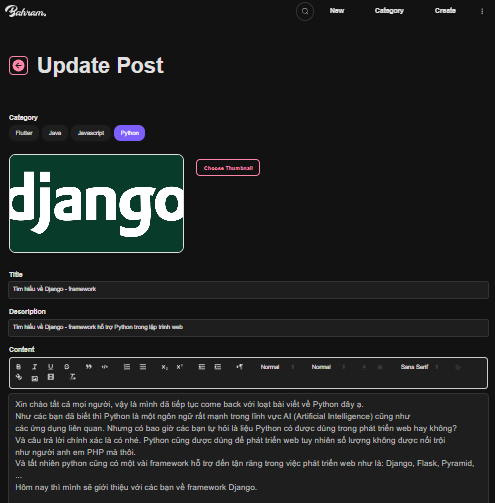
### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:**Chức năng cho phép người dùng chỉnh sửa các bài viết đã đăng để cập nhật thông tin hoặc sửa lỗi chính tả, định dạng, v.v.

**Bước sử dụng:**

1. Truy cập bài viết bạn đã đăng
2. Nhấn nút “Chỉnh sửa”
3. Cập nhật tiêu đề, nội dung hoặc hình ảnh
4. Nhấn “Lưu thay đổi”
5. Hệ thống xác nhận cập nhật thành công

**Hình ảnh minh họa:**



Hình 27. Cập nhật bài viết

**Tooltip & hỗ trợ:**

* Tooltip cho ảnh: “Tải ảnh mới nếu muốn thay đổi ảnh hiện tại”
* Hiển thị thời gian cập nhật cuối cùng bên dưới bài viết

## US05 – Xóa bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US05

**Tên chức năng:** Xóa bài viết

**Mục đích:** Cho phép người dùng chủ động xóa các bài viết không còn phù hợp hoặc muốn loại bỏ khỏi hệ thống.

**Câu chuyện người dùng:**

* Là người dùng, tôi muốn xóa một bài viết để loại bỏ nội dung không cần thiết.

**Acceptance Criteria:**

* Hiển thị hộp thoại xác nhận khi người dùng chọn “Xóa”.
* Chỉ cho phép xóa bài viết do chính người dùng tạo.
* Sau khi xác nhận, bài viết được xóa khỏi hệ thống (hoặc ẩn nếu dùng soft delete).
* Không hiển thị bài viết đã xóa trong danh sách công khai hoặc trang cá nhân.
* Trả về thông báo kết quả thành công hoặc lỗi.

**Độ phức tạp (Story Point):** 10  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (kiểm tra quyền xóa và xác thực)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Nút “Xóa” trên giao diện bài viết
* Hộp thoại xác nhận: “Bạn có chắc chắn muốn xóa bài viết này không?”
* Toast/Alert thông báo kết quả

**Giao diện đầu vào/đầu ra:**

* **Đầu vào:**
* postId: string (ID bài viết cần xóa)
* Token xác thực người dùng (Bearer Token)
* **Đầu ra:**
* 200 OK nếu xóa thành công
* 403 Forbidden nếu người dùng không có quyền xóa
* 404 Not Found nếu bài viết không tồn tại

**Ghi chú kỹ thuật:**

* Có thể sử dụng **soft delete** để giữ dữ liệu (thêm trường isDeleted: boolean)
* Nếu dùng hard delete, cần xóa hình ảnh và liên kết liên quan
* Xóa quyền truy cập và phản hồi liên quan đến bài viết (nếu cần)

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:**Chức năng giúp người dùng xóa bài viết đã đăng nếu không còn phù hợp hoặc muốn loại bỏ khỏi hệ thống.

**Bước sử dụng:**

1. Truy cập bài viết của bạn
2. Nhấn nút “Xóa”
3. Hộp thoại hiện ra → xác nhận hành động
4. Sau khi xóa, hệ thống thông báo thành công và ẩn bài viết khỏi hệ thống

**Hình ảnh minh họa:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 28. Xoá bài viết

**Tooltip & hỗ trợ:**

* Tooltip nút “Xóa”: “Loại bỏ bài viết khỏi blog của bạn”
* Cảnh báo: “Thao tác này không thể hoàn tác!” (nếu là hard delete)

## US06 – Phân loại bài viết theo danh mục

### 1.1. Mô tả User Story

**Mã User Story:** US06

**Tên chức năng:** Phân loại bài viết theo danh mục

**Mục đích:** Giúp người dùng tổ chức và quản lý các bài viết của mình một cách khoa học, dễ tìm kiếm và truy cập thông qua danh mục.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn phân loại bài viết theo danh mục để tổ chức nội dung một cách khoa học.*

**Acceptance Criteria:**

* Cho phép người dùng tạo danh mục mới.
* Cho phép sửa tên hoặc xóa danh mục đã tạo.
* Gán một hoặc nhiều bài viết vào một danh mục cụ thể.
* Lọc danh sách bài viết theo danh mục.
* Không cho phép xóa danh mục nếu còn bài viết đang gán (hoặc yêu cầu chuyển/cảnh báo).
* Người dùng chỉ nhìn thấy danh mục và bài viết của chính mình.

**Độ phức tạp (Story Point):** 12  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (quyền sở hữu danh mục, bài viết)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

Giao diện quản lý danh mục: danh sách + nút “Thêm danh mục”

* Giao diện tạo/sửa bài viết: chọn danh mục từ dropdown/multi-select
* Giao diện lọc bài viết theo danh mục

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

* categoryName: string
* postId: string
* categoryId: string

**Đầu ra:**

* 200 OK khi tạo, cập nhật, xóa, gán thành công
* 400 Bad Request nếu dữ liệu thiếu hoặc sai
* 403 Forbidden nếu thao tác trên danh mục không thuộc quyền

**Ghi chú kỹ thuật:**

* Nếu sử dụng danh mục toàn cục → cần kiểm tra quyền sử dụng
* Có thể dùng Enum cho danh mục phổ biến (nếu cố định)
* Cần cập nhật lại bài viết khi danh mục bị thay đổi

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng tạo các danh mục riêng để nhóm bài viết theo chủ đề như “Công nghệ”, “Du lịch”, “Sức khỏe”,...

**Bước sử dụng:**

1. Vào trang quản lý danh mục → nhấn “Thêm danh mục”
2. Tạo/sửa bài viết → chọn danh mục tương ứng từ danh sách
3. Truy cập danh sách bài viết → sử dụng bộ lọc danh mục để lọc nhanh

**Hình ảnh minh họa:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 29. Phân loại bài viết theo danh mục

**Hình ảnh phân loại bài viết theo danh mục**

**Tooltip & hỗ trợ:**

* Tooltip “Chỉ người tạo danh mục mới có thể sửa hoặc xóa”
* Tooltip lọc: “Lọc nhanh bài viết theo chủ đề”

## US07 – Lọc và sắp xếp bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US07

**Tên chức năng:** Lọc và sắp xếp bài viết

**Mục đích:** Cho phép người dùng dễ dàng tìm kiếm, quản lý và truy cập các bài viết thông qua các tiêu chí lọc và sắp xếp như danh mục hoặc ngày đăng.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn lọc và sắp xếp bài viết theo danh mục hoặc ngày tháng để dễ dàng quản lý nội dung.*

**Acceptance Criteria:**

* Cho phép lọc bài viết theo danh mục đã chọn
* Cho phép lọc bài viết theo khoảng thời gian (ngày/tháng/năm)
* Cho phép sắp xếp bài viết theo: ngày đăng (mới nhất/cũ nhất), tiêu đề (A-Z/Z-A)
* Hiển thị đúng danh sách bài viết theo điều kiện lọc và sắp xếp
* Lưu trạng thái lọc/sắp xếp khi quay lại trang

**Độ phức tạp (Story Point):** 10  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** Yes (chỉ lọc/sắp xếp bài viết thuộc người dùng)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Bộ lọc nằm phía trên danh sách bài viết, gồm dropdown danh mục, date picker, và combo sắp xếp
* Kết quả hiển thị dạng danh sách với thông tin bài viết cơ bản

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

filter = {

  categoryId?: string,

  startDate?: string,

  endDate?: string,

  sortBy?: 'createdAt' | 'title',

  sortOrder?: 'asc' | 'desc',

}

**Đầu ra:**

* 200 OK + danh sách bài viết đã lọc và sắp xếp
* 400 Bad Request nếu thông tin lọc không hợp lệ

**Ghi chú kỹ thuật:**

* Có thể dùng query param hoặc body (nếu POST cho tìm nâng cao)
* Pagination nên được kết hợp với lọc
* Các field sắp xếp cần được đánh chỉ mục (indexed)

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng tìm kiếm, lọc và sắp xếp các bài viết đã tạo một cách linh hoạt và trực quan.

**Bước sử dụng:**

1. Truy cập trang danh sách bài viết
2. Chọn danh mục từ dropdown (tùy chọn)
3. Chọn bài viết phổ biến, gần đây hoặc tất cả
4. Xem kết quả được cập nhật ngay lập tức

**Hình ảnh minh họa:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 30. Lọc và sắp xếp bài viết

**Tooltip & hỗ trợ:**

* Tooltip "Chọn danh mục để lọc bài viết theo gần đây, phổ biến hoặc tất cả"

## US08 – Lên lịch đăng bài tự động

### 1.1. Mô tả User Story

**Mã User Story:** US08

**Tên chức năng:** Lên lịch đăng bài tự động

**Mục đích:** Giúp người dùng chủ động lên lịch đăng bài viết trong tương lai, tối ưu hóa thời gian và hiệu quả tiếp cận người đọc.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn lên lịch đăng bài tự động để tối ưu hóa thời gian đăng bài.*

**Acceptance Criteria:**

* Cho phép chọn ngày và giờ cụ thể để đăng bài
* Bài viết sẽ được đăng tự động đúng thời gian đã lên lịch
* Có thông báo cho người dùng khi bài viết được đăng thành công
* Cho phép chỉnh sửa hoặc hủy lịch đăng nếu bài chưa được đăng
* Trạng thái bài viết thay đổi từ “Đã lên lịch” → “Đã đăng” sau khi đăng

**Độ phức tạp (Story Point):** 20  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (chỉ người tạo mới được chỉnh sửa lịch đăng)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Form tạo/chỉnh sửa bài viết có thêm mục “Lên lịch đăng bài” với DateTime picker
* Màn hình quản lý bài viết hiển thị trạng thái: Đã đăng / Đã lên lịch

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

{

  title: "string",

  content: "string",

  scheduledAt: "2025-06-10T10:00:00Z"  // ISO 8601

}

**Đầu ra:**

* 200 OK khi lịch đăng được tạo/thay đổi thành công
* 400 Bad Request nếu ngày giờ không hợp lệ (trong quá khứ)
* 403 Forbidden nếu người dùng không phải là tác giả

**Ghi chú kỹ thuật:**

* Job scheduler backend sử dụng cron hoặc queue (Bull, Agenda, Celery)
* Server xử lý job đăng đúng thời gian, cập nhật status và gửi thông báo
* Cần xử lý timezone của người dùng

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng cho phép người dùng hẹn giờ đăng bài viết trong tương lai, không cần thao tác thủ công lúc đăng.

**Bước sử dụng:**

1. Vào trang tạo hoặc chỉnh sửa bài viết
2. Nhập tiêu đề và nội dung
3. Bật “Lên lịch đăng bài” và chọn ngày giờ mong muốn
4. Nhấn “Lưu và lên lịch”
5. Bài viết sẽ được đăng tự động đúng thời gian

**Tooltip & hỗ trợ:**

* Tooltip: “Bài viết sẽ được tự động đăng vào thời gian đã chọn”
* Thông báo qua UI hoặc email khi đăng thành công

## US09 – Quản lý hình ảnh và tệp đính kèm

### 1.1. Mô tả User Story

**Mã User Story:** US09

**Tên chức năng:** Quản lý hình ảnh và tệp đính kèm

**Mục đích:** Cho phép người dùng tải lên, chỉnh sửa và quản lý hình ảnh/tệp đính kèm để chèn vào bài viết, nâng cao tính trực quan và minh họa nội dung.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn quản lý hình ảnh và tệp đính kèm để minh họa cho bài viết.*

**Acceptance Criteria:**

* Cho phép tải lên hình ảnh và các tệp đính kèm phổ biến (PDF, DOCX, ZIP…)
* Hình ảnh được hiển thị trong bài viết tại vị trí chèn
* Có thể xoá hoặc cập nhật hình ảnh/tệp đã tải
* Hiển thị thư viện media để tái sử dụng hình ảnh đã đăng
* Giới hạn dung lượng và định dạng phù hợp
* Có thông báo khi thao tác tải lên thành công/thất bại

**Độ phức tạp (Story Point):** 10  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (chỉ tác giả mới được chỉnh sửa, xóa tệp của mình)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Giao diện soạn bài viết có nút "Tải hình ảnh" hoặc "Thêm tệp đính kèm"
* Thư viện ảnh dạng lưới, cho phép xem trước, xóa, chèn

**Giao diện đầu vào/đầu ra:**

**Đầu vào (form upload):**

* file: hình ảnh (.jpg, .png, .gif), tài liệu (.pdf, .docx), zip
* description (optional)

**Đầu ra:**

* 201 Created với URL truy cập file
* 400 Bad Request nếu định dạng không hỗ trợ
* 413 Payload Too Large nếu vượt dung lượng cho phép

**Cấu trúc dữ liệu liên quan:**

**Ghi chú kỹ thuật:**

* Lưu file trên cloud (S3, Firebase Storage, hoặc local dev mode)
* Resize ảnh trước khi lưu nếu quá lớn
* Quét virus/tệp độc hại nếu có (ClamAV)

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng hỗ trợ người dùng chèn và quản lý hình ảnh, tài liệu minh họa vào bài viết một cách trực quan và dễ dàng.

**Bước sử dụng:**

1. Khi tạo hoặc chỉnh sửa bài viết, chọn "Tải ảnh/tệp"
2. Chọn file từ máy và nhấn “Tải lên”
3. Hệ thống tự động hiển thị trong thư viện đính kèm
4. Nhấn vào ảnh để chèn vào nội dung
5. Có thể xóa, đổi ảnh bất kỳ lúc nào

**Tooltip & hỗ trợ:**

* Tooltip: “Chỉ hỗ trợ ảnh JPG/PNG/GIF, tệp PDF/DOCX dưới 10MB”
* Thông báo lỗi cụ thể nếu tệp bị từ chối

## US10 – Tìm kiếm bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US10

**Tên chức năng:** Tìm kiếm bài viết

**Mục đích:** Cho phép người dùng tìm kiếm bài viết theo từ khóa, tiêu đề hoặc nội dung để nhanh chóng tìm thấy thông tin cần thiết.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn tìm kiếm bài viết theo từ khóa, tiêu đề hoặc nội dung để dễ dàng tìm thấy bài viết cần thiết.*

**Acceptance Criteria:**

* Người dùng nhập từ khóa tìm kiếm (có thể là từ trong tiêu đề hoặc nội dung bài viết)
* Hệ thống trả về danh sách bài viết phù hợp với từ khóa
* Kết quả được sắp xếp theo độ liên quan hoặc thời gian mới nhất
* Hỗ trợ tìm kiếm không phân biệt hoa/thường, tìm gần đúng
* Hiển thị thông báo nếu không tìm thấy bài viết phù hợp

**Độ phức tạp (Story Point):** 12  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (chỉ hiện bài viết người dùng được quyền xem)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Thanh tìm kiếm trên đầu trang hoặc trong trang quản lý bài viết
* Kết quả tìm kiếm hiển thị dạng danh sách với tiêu đề, đoạn trích nội dung, ngày tạo

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

* query: string (từ khóa tìm kiếm)

**Đầu ra:**

* 200 OK với danh sách bài viết thỏa mãn
* 204 No Content nếu không có kết quả

**Ghi chú kỹ thuật:**

* Sử dụng ElasticSearch hoặc full-text search trên DB (PostgreSQL, MySQL)
* Áp dụng stemming, stop words để tăng chất lượng tìm kiếm
* Phân trang kết quả nếu nhiều bài viết
* Kiểm tra quyền truy cập bài viết theo user

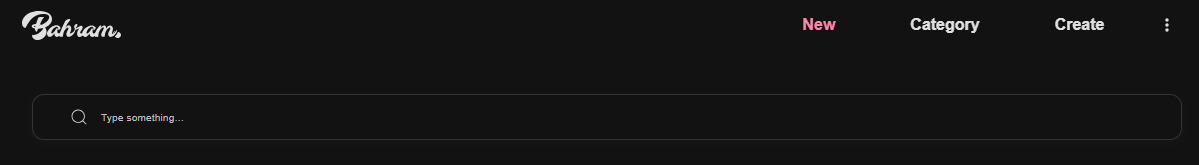
### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng tìm kiếm bài viết nhanh chóng bằng cách nhập từ khóa liên quan đến tiêu đề hoặc nội dung bài viết.

**Bước sử dụng:**

1. Truy cập trang quản lý bài viết hoặc thanh tìm kiếm chính
2. Nhập từ khóa vào ô tìm kiếm
3. Nhấn Enter hoặc nút tìm kiếm
4. Hệ thống hiển thị danh sách bài viết liên quan, có thể chọn bài để xem/chỉnh sửa
5. Nếu không tìm thấy, hiển thị thông báo tương ứng

**Hình ảnh minh họa:**



Hình 31. Tìm kiếm bài viết

**Tooltip & hỗ trợ:**

* Tooltip: “Nhập từ khóa tìm kiếm bài viết theo tiêu đề hoặc nội dung”
* Hiển thị loading khi đang tìm kiếm

## US11 – Tối ưu hóa SEO tự động

### 1.1. Mô tả User Story

**Mã User Story:** US11

**Tên chức năng:** Tối ưu hóa SEO tự động

**Mục đích:** Giúp người dùng tối ưu hóa nội dung bài viết một cách tự động nhằm tăng khả năng hiển thị trên các công cụ tìm kiếm.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn hệ thống tự động tối ưu hóa nội dung cho SEO để tăng khả năng hiển thị trên các công cụ tìm kiếm.*

**Acceptance Criteria:**

* Người dùng nhập hoặc chọn bài viết cần tối ưu SEO
* Hệ thống phân tích từ khóa chính và phụ trong bài viết
* Đề xuất cải thiện các yếu tố SEO như tiêu đề, meta description, thẻ heading, mật độ từ khóa
* Hệ thống tự động tối ưu và cập nhật tiêu đề, meta description, cấu trúc bài viết theo chuẩn SEO
* Người dùng có thể xem và chỉnh sửa các đề xuất trước khi lưu
* Hệ thống lưu lại phiên bản tối ưu SEO của bài viết
* Hiển thị thông báo khi quá trình tối ưu hoàn tất

**Độ phức tạp (Story Point):** 22  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes (bảo mật nội dung và dữ liệu người dùng)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Giao diện phân tích SEO bên trong trang chỉnh sửa bài viết hoặc trang quản lý SEO
* Các đề xuất SEO hiển thị dạng bảng hoặc khung tooltip kèm nút chấp nhận/chỉnh sửa
* Nút "Tối ưu tự động" cho phép hệ thống cập nhật nhanh các đề xuất

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

* postId: string (ID bài viết cần tối ưu SEO)
* content: string (nội dung bài viết hiện tại)

**Đầu ra:**

* Đề xuất SEO:
  + optimizedTitle: string
  + metaDescription: string
  + headings: array (đề xuất sửa thẻ H1, H2, ...)
  + keywordDensity: object (tỷ lệ xuất hiện các từ khóa)
  + seoScore: number (điểm đánh giá SEO)
* Trạng thái cập nhật: success/failure

**Ghi chú kỹ thuật:**

* Sử dụng AI/ML hoặc thuật toán phân tích ngôn ngữ tự nhiên (NLP) để phân tích và đề xuất
* Áp dụng các quy tắc SEO chuẩn (Google SEO guidelines)
* Có thể tích hợp công cụ bên thứ ba như Yoast SEO API, SEMrush API
* Cập nhật dữ liệu bài viết trong DB sau khi tối ưu
* Framework đề xuất: Python (Flask/Django) cho backend xử lý AI, React/Vue cho frontend

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng tự động tối ưu hóa nội dung bài viết theo chuẩn SEO, tăng khả năng tiếp cận trên các công cụ tìm kiếm.

**Bước sử dụng:**

1. Mở bài viết cần tối ưu trong trang quản lý hoặc chỉnh sửa bài viết
2. Nhấn nút "Tối ưu SEO tự động"
3. Xem các đề xuất được hệ thống phân tích và hiển thị
4. Chỉnh sửa đề xuất nếu cần hoặc chấp nhận tự động cập nhật
5. Lưu bài viết sau khi tối ưu thành công
6. Nhận thông báo xác nhận quá trình hoàn tất

**Tooltip & hỗ trợ:**

* Tooltip: “Tối ưu tiêu đề, meta description và cấu trúc bài viết để đạt chuẩn SEO”
* Hiển thị cảnh báo nếu bài viết có lỗi SEO nghiêm trọng

## US12 – Thông báo khi đăng bài thành công

### 1.1. Mô tả User Story

**Mã User Story:** US12

**Tên chức năng:** Thông báo khi đăng bài thành công

**Mục đích:** Giúp người dùng nhận thông báo ngay khi bài viết được đăng thành công để dễ dàng theo dõi trạng thái đăng bài.

**Câu chuyện người dùng:** *Là người dùng, tôi muốn nhận thông báo khi bài viết được đăng thành công để theo dõi trạng thái đăng bài.*

**Acceptance Criteria:**

* Khi bài viết được đăng hoặc lên lịch đăng thành công, hệ thống gửi thông báo đến người dùng
* Thông báo được gửi qua email và/hoặc thông báo trong hệ thống (in-app notification)
* Nội dung thông báo bao gồm tiêu đề bài viết, thời gian đăng, trạng thái thành công
* Người dùng có thể xem lịch sử thông báo trong phần quản lý thông báo
* Hệ thống xử lý lỗi gửi thông báo và ghi nhận lại nếu gửi thất bại

**Độ phức tạp (Story Point):** 8  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** Yes (bảo mật thông tin người dùng và nội dung thông báo)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Giao diện phần thông báo trong trang quản lý cá nhân hoặc dashboard
* Popup hoặc toast notification khi đăng bài thành công
* Email mẫu gửi thông báo đến người dùng

**Giao diện đầu vào/đầu ra:**

**Đầu vào:**

* postId: string (ID bài viết)
* userId: string (ID người dùng)
* postTitle: string (Tiêu đề bài viết)
* publishTime: datetime (Thời gian bài viết được đăng)

**Đầu ra:**

* Thông báo được gửi thành công hoặc lỗi
* Thông báo lưu trong hệ thống với trạng thái đọc/chưa đọc

**Ghi chú kỹ thuật:**

* Sử dụng dịch vụ gửi email (SMTP, SendGrid, Amazon SES, v.v.)
* Sử dụng WebSocket hoặc Push Notification để gửi thông báo realtime trong app
* Lưu trữ thông báo trong bảng notifications cho lịch sử
* Đảm bảo bảo mật dữ liệu và chống spam thông báo
* Framework đề xuất: Node.js/Python cho backend, React/Flutter cho frontend

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng nhận được thông báo ngay khi bài viết được đăng thành công qua email và trong ứng dụng.

**Bước sử dụng:**

1. Đăng nhập vào hệ thống và tạo bài viết mới hoặc chỉnh sửa bài viết có lịch đăng
2. Khi bài viết được đăng thành công, người dùng sẽ nhận được thông báo qua email và/hoặc trong app
3. Truy cập phần “Thông báo” để xem lịch sử các thông báo đã nhận
4. Mở email để xem chi tiết thông báo đăng bài

**Hình ảnh minh họa:**

A green rectangle with white text

AI-generated content may be incorrect.

Hình 32. Thông báo đăng bài thành công

**Tooltip & hỗ trợ:**

* Tooltip tại nút đăng bài: “Bạn sẽ nhận thông báo khi bài viết được đăng thành công”
* Hỗ trợ kiểm tra email nhận thông báo

## US13 – Thống kê hiệu suất bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US13  
**Tên chức năng:** Thống kê hiệu suất bài viết  
**Mục đích:** Cung cấp dữ liệu phân tích để người dùng đánh giá hiệu quả nội dung thông qua các chỉ số như lượt xem, thời gian đọc, tỷ lệ tương tác, và biểu đồ hiệu suất theo thời gian.

**Câu chuyện người dùng:** Là người dùng, tôi muốn xem thống kê hiệu suất của bài viết để đánh giá hiệu quả của chiến lược nội dung.

**Acceptance Criteria:**

* Hiển thị tổng số lượt xem bài viết theo thời gian
* Hiển thị thời gian trung bình người dùng đọc bài viết
* Tính và hiển thị tỷ lệ tương tác (reaction, comment, share)
* Vẽ biểu đồ hiệu suất bài viết (xem theo ngày/tuần/tháng)
* Có thể lọc theo khoảng thời gian cụ thể
* Giao diện phản hồi tốt khi không có dữ liệu
* Chỉ người viết bài (hoặc admin) mới xem được thống kê

**Độ phức tạp (Story Point):** 18  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** Yes (bảo mật dữ liệu thống kê và quyền truy cập)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Trang chi tiết thống kê bài viết với dashboard tổng quan
* Panel hiển thị các số liệu chính: lượt xem, thời gian đọc TB, tương tác
* Biểu đồ dạng line/bar thể hiện xu hướng theo thời gian
* Bộ lọc thời gian với date picker

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* postId: string (ID bài viết)
* userId: string (ID người dùng)
* fromDate: datetime (Ngày bắt đầu lọc - optional)
* toDate: datetime (Ngày kết thúc lọc - optional)

*Đầu ra:*

* Dữ liệu thống kê chi tiết được hiển thị trên dashboard
* Biểu đồ hiệu suất được render thành công
* Thông báo lỗi nếu không có quyền truy cập hoặc dữ liệu không tồn tại

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người viết bài theo dõi và phân tích hiệu suất của từng bài viết thông qua các chỉ số quan trọng và biểu đồ trực quan.

**Bước sử dụng:**

1. Đăng nhập vào hệ thống và truy cập danh sách bài viết của bạn
2. Nhấn vào nút "Thống kê" hoặc biểu tượng analytics bên cạnh bài viết
3. Xem tổng quan các chỉ số: lượt xem, thời gian đọc, tỷ lệ tương tác
4. Sử dụng bộ lọc thời gian để xem dữ liệu trong khoảng thời gian cụ thể
5. Phân tích biểu đồ xu hướng để đánh giá hiệu suất bài viết theo thời gian

**Tooltip & hỗ trợ:**

* Tooltip tại tỷ lệ tương tác: "Tỷ lệ tương tác = (likes + comments + shares) / tổng lượt xem × 100%"
* Hỗ trợ giải thích các chỉ số: "Thời gian đọc trung bình được tính dựa trên thời gian người dùng ở lại trang bài viết"

**Ghi chú kỹ thuật:**

* Sử dụng caching (Redis) để tối ưu hóa truy vấn dữ liệu thống kê
* Sử dụng thư viện biểu đồ (Chart.js, D3.js) để render visualization
* Lưu trữ dữ liệu thống kê trong bảng post\_statistics
* Đảm bảo bảo mật quyền truy cập và chống truy cập trái phép
* Framework đề xuất: Node.js/Python cho backend, React/Vue.js cho frontend

## US14 – Chia sẻ bài viết lên mạng xã hội

### 1.1. Mô tả User Story

**Mã User Story:** US14  
**Tên chức năng:** Chia sẻ bài viết lên mạng xã hội  
**Mục đích:** Giúp người dùng chia sẻ bài viết lên các nền tảng mạng xã hội để tăng khả năng tiếp cận nội dung.

**Câu chuyện người dùng:** Là người dùng, tôi muốn chia sẻ bài viết lên các nền tảng mạng xã hội trực tiếp từ hệ thống để tăng khả năng tiếp cận nội dung của tôi.

**Acceptance Criteria:**

* Có nút chia sẻ bài viết với các biểu tượng mạng xã hội (Facebook, Twitter, LinkedIn...)
* Khi nhấn nút chia sẻ, hệ thống mở liên kết chia sẻ tương ứng
* Nội dung chia sẻ bao gồm tiêu đề và đường dẫn bài viết
* Giao diện chia sẻ hiển thị tốt cả mobile và desktop

**Độ phức tạp (Story Point):** 5  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** No

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Khu vực chia sẻ với các icon mạng xã hội ở cuối bài viết
* Tooltip hướng dẫn khi hover vào các nút chia sẻ
* Giao diện responsive cho mobile và desktop

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* postId: string (ID bài viết)
* postTitle: string (Tiêu đề bài viết)
* postUrl: string (URL bài viết)
* platform: string (Tên nền tảng mạng xã hội)

*Đầu ra:*

* Liên kết chia sẻ được mở trong tab mới
* Nội dung chia sẻ được format đúng cho từng nền tảng

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng giúp người dùng dễ dàng chia sẻ bài viết lên các nền tảng mạng xã hội phổ biến chỉ với một cú click.

**Bước sử dụng:**

1. Đọc đến cuối bài viết để tìm khu vực chia sẻ
2. Nhấn vào biểu tượng của nền tảng mạng xã hội bạn muốn chia sẻ
3. Hệ thống sẽ mở tab mới với nội dung chia sẻ đã được điền sẵn
4. Xác nhận chia sẻ trên nền tảng đó

**Hình ảnh minh họa:**

A screenshot of a social media post

AI-generated content may be incorrect.

Hình 33. Chia sẻ bài viết lên mạng xã hội

**Tooltip & hỗ trợ:**

* Tooltip: "Chia sẻ bài viết này lên [Tên nền tảng]"
* Hỗ trợ: "Bạn có thể nhấn chuột phải → 'Sao chép liên kết' để chia sẻ thủ công"

**Ghi chú kỹ thuật:**

* Sử dụng SDK hoặc API chia sẻ của các nền tảng (Facebook SDK, Twitter Web Intent...)
* URL bài viết sinh tự động theo slug/title
* UI responsive bằng Tailwind CSS
* Framework đề xuất: JavaScript vanilla hoặc React cho frontend

## US15 – Giao diện thân thiện đa thiết bị

### 1.1. Mô tả User Story

**Mã User Story:** US15  
**Tên chức năng:** Giao diện thân thiện đa thiết bị  
**Mục đích:** Đảm bảo người dùng có trải nghiệm thoải mái trên mọi loại thiết bị.

**Câu chuyện người dùng:** Là người dùng, tôi muốn trải nghiệm hệ thống một cách thoải mái trên mọi loại thiết bị để tôi có thể sử dụng nó trên mobile, tablet, hoặc desktop một cách dễ dàng.

**Acceptance Criteria:**

* Giao diện responsive hiển thị tốt trên các màn hình khác nhau
* Không cần cuộn ngang, nội dung co giãn hợp lý trên mobile
* Các nút và input đủ lớn để dễ thao tác bằng cảm ứng
* Menu điều hướng rõ ràng có thể đóng/mở trên mobile

**Độ phức tạp (Story Point):** 13  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** No

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Layout responsive với breakpoint cho desktop, tablet, mobile
* Menu hamburger cho mobile
* Grid system co giãn theo kích thước màn hình

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* screenWidth: number (Chiều rộng màn hình)
* deviceType: string (desktop/tablet/mobile)
* userAgent: string (Thông tin thiết bị)

*Đầu ra:*

* Layout được điều chỉnh phù hợp với thiết bị
* Menu và navigation được tối ưu hóa
* Kích thước font và button phù hợp

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Hệ thống tự động điều chỉnh giao diện để phù hợp với mọi thiết bị, đảm bảo trải nghiệm tối ưu.

**Bước sử dụng:**

1. Truy cập hệ thống từ bất kỳ thiết bị nào
2. Giao diện sẽ tự động điều chỉnh theo kích thước màn hình
3. Sử dụng menu hamburger trên mobile để điều hướng
4. Tất cả chức năng đều có thể truy cập dễ dàng

**Hình ảnh minh họa:**

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 34. Giao diện trên thiết bị mobile

A screenshot of a computer

AI-generated content may be incorrect.

Hình 35. Giao diện trên desktop

**Tooltip & hỗ trợ:**

* Hỗ trợ: "Hệ thống hỗ trợ mọi kích thước màn hình từ 320px đến 1920px+"

**Ghi chú kỹ thuật:**

* Sử dụng TailwindCSS với breakpoint: sm, md, lg, xl
* Ưu tiên thiết kế Mobile-first
* Menu chuyển sang hamburger icon khi mobile
* Framework đề xuất: React/Vue.js với TailwindCSS

## US16 – Chế độ xem trước bài viết

### 1.1. Mô tả User Story

**Mã User Story:** US16  
**Tên chức năng:** Chế độ xem trước bài viết  
 **Mục đích:** Giúp người dùng xem trước bài viết trước khi đăng để đảm bảo mọi thứ hiển thị đúng.

**Câu chuyện người dùng:** Là người dùng, tôi muốn xem trước bài viết của mình trước khi đăng để đảm bảo mọi thứ trông ổn trước khi xuất bản.

**Acceptance Criteria:**

* Có nút "Xem trước" hiển thị bản xem trước bài viết
* Hiển thị giống bài viết thật bao gồm định dạng, ảnh, liên kết
* Có thể quay lại chỉnh sửa sau khi xem trước
* Preview responsive trên mọi thiết bị

**Độ phức tạp (Story Point):** 8  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** No

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Button "Xem trước" trong editor
* Modal hoặc tab riêng hiển thị preview
* Navigation để chuyển đổi giữa edit và preview mode

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* postContent: string (Nội dung bài viết)
* postTitle: string (Tiêu đề bài viết)
* postImages: array (Danh sách hình ảnh)
* postMetadata: object (Metadata bài viết)

*Đầu ra:*

* HTML preview được render
* Layout giống bài viết thực tế
* Khả năng quay lại chỉnh sửa

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng xem trước giúp bạn kiểm tra bài viết sẽ hiển thị như thế nào trước khi xuất bản chính thức.

**Bước sử dụng:**

1. Soạn thảo nội dung bài viết trong editor
2. Nhấn nút "Xem trước"
3. Kiểm tra định dạng, hình ảnh, liên kết trong preview
4. Quay lại chỉnh sửa nếu cần hoặc tiến hành đăng bài

**Hình ảnh minh họa:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 36. Chế độ xem trước bài viết

**Tooltip & hỗ trợ:**

* Tooltip: "Xem bài viết sẽ hiển thị như thế nào khi được đăng"
* Hỗ trợ: "Preview sẽ hiển thị chính xác như bài viết thực tế"

**Ghi chú kỹ thuật:**

* Render preview bằng cách reuse component hiển thị bài viết
* Modal hoặc section riêng cho preview
* Dùng Markdown hoặc HTML preview engine
* Framework đề xuất: React/Vue.js với markdown parser

## US17 – Nhập/Xuất dữ liệu

### 1.1. Mô tả User Story

**Mã User Story:** US17  
**Tên chức năng:** Nhập/Xuất dữ liệu  
**Mục đích:** Giúp người dùng sao lưu hoặc chuyển nội dung một cách dễ dàng.

**Câu chuyện người dùng:** Là người dùng, tôi muốn nhập và xuất dữ liệu để có thể sao lưu hoặc chuyển nội dung một cách dễ dàng.

**Acceptance Criteria:**

* Có nút "Nhập" và "Xuất" dữ liệu hiển thị rõ ràng trên giao diện
* Xuất dữ liệu thành công tạo file định dạng CSV, PDF hoặc Word
* Nhập dữ liệu hợp lệ, hệ thống đọc và hiển thị đúng nội dung
* Nhập dữ liệu không hợp lệ hiển thị thông báo lỗi rõ ràng

**Độ phức tạp (Story Point):** 13  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** Yes (bảo mật dữ liệu người dùng)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Buttons "Nhập dữ liệu" và "Xuất dữ liệu" trong settings
* Modal chọn định dạng file khi xuất
* File upload interface khi nhập dữ liệu

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* userId: string (ID người dùng)
* exportFormat: string (csv/pdf/docx)
* importFile: file (File được upload)
* dataType: string (posts/users/settings)

*Đầu ra:*

* File được tạo và download khi xuất
* Dữ liệu được import và hiển thị khi nhập
* Thông báo lỗi nếu file không hợp lệ

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Chức năng nhập/xuất giúp bạn sao lưu dữ liệu hoặc chuyển nội dung giữa các hệ thống.

**Bước sử dụng:**

*Để xuất dữ liệu:*

1. Truy cập phiên "Cài đặt" > "Dữ liệu"
2. Nhấn nút "Xuất dữ liệu"
3. Chọn định dạng file mong muốn (CSV/PDF/Word)
4. File sẽ được tải về máy tự động

*Để nhập dữ liệu:*

1. Nhấn nút "Nhập dữ liệu"
2. Chọn file từ máy tính
3. Hệ thống sẽ xác thực và hiển thị dữ liệu
4. Xác nhận để hoàn tất việc nhập

**Tooltip & hỗ trợ:**

* Tooltip: "Xuất tất cả dữ liệu của bạn để sao lưu"
* Hỗ trợ: "Đảm bảo file nhập có cấu trúc đúng để tránh lỗi"

**Ghi chú kỹ thuật:**

* Sử dụng thư viện như FileSaver.js để hỗ trợ xuất file
* Kiểm tra định dạng và cấu trúc file khi nhập dữ liệu
* Dữ liệu được xử lý và validate trước khi lưu
* Framework đề xuất: Node.js với multer cho upload, jsPDF cho export

## US18 – Bảo mật hệ thống

### 1.1. Mô tả User Story

**Mã User Story:** US18  
**Tên chức năng:** Bảo mật hệ thống  
**Mục đích:** Đảm bảo an toàn cho dữ liệu và quyền riêng tư của người dùng.

**Câu chuyện người dùng:** Là người dùng, tôi muốn hệ thống đảm bảo bảo mật để dữ liệu và quyền riêng tư của tôi được bảo vệ.

**Acceptance Criteria:**

* Xác thực hai lớp (2FA) qua email hoặc SMS
* Mã hóa dữ liệu nhạy cảm khi lưu trữ
* Bảo vệ khỏi các cuộc tấn công phổ biến như XSS, CSRF
* Đăng xuất tự động sau thời gian không hoạt động

**Độ phức tạp (Story Point):** 21  
**Độ ưu tiên:** Cao  
**Có liên quan bảo mật:** Yes

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Form đăng nhập với tùy chọn 2FA
* Settings bảo mật với các tùy chọn cấu hình
* Thông báo cảnh báo bảo mật khi phát hiện hoạt động bất thường

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* username: string (Tên đăng nhập)
* password: string (Mật khẩu)
* twoFactorCode: string (Mã 2FA)
* sessionTimeout: number (Thời gian hết phiên)

*Đầu ra:*

* JWT token được mã hóa
* Session được thiết lập với thời gian hết hạn
* Thông báo bảo mật khi cần thiết

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Hệ thống bảo mật đa lớp giúp bảo vệ tài khoản và dữ liệu của bạn khỏi các mối đe dọa.

**Bước sử dụng:**

*Để kích hoạt 2FA:*

1. Truy cập "Cài đặt" > "Bảo mật"
2. Bật tùy chọn "Xác thực hai lớp"
3. Chọn phương thức: Email hoặc SMS
4. Nhập mã xác thực để hoàn tất

*Đăng nhập an toàn:*

1. Nhập thông tin tài khoản
2. Nhập mã xác thực được gửi qua email/SMS
3. Hệ thống sẽ tự động đăng xuất sau 15 phút không hoạt động

**Tooltip & hỗ trợ:**

* Tooltip: "2FA giúp bảo vệ tài khoản ngay cả khi mật khẩu bị lộ"
* Hỗ trợ: "Không chia sẻ mã xác thực 2FA với bất kỳ ai"

**Ghi chú kỹ thuật:**

* Sử dụng bcrypt hoặc argon2 để mã hóa mật khẩu
* Triển khai 2FA qua email hoặc SMS
* Sử dụng Content Security Policy để chống XSS
* Thiết lập thời gian hết hạn phiên làm việc
* Framework đề xuất: Node.js với passport.js, helmet.js cho bảo mật

## US19 – Sao lưu tự động

### 1.1. Mô tả User Story

**Mã User Story:** US19  
**Tên chức năng:** Sao lưu tự động  
**Mục đích:** Giúp người dùng tránh mất thông tin quan trọng thông qua việc sao lưu tự động.

**Câu chuyện người dùng:** Là người dùng, tôi muốn hệ thống tự động sao lưu dữ liệu của tôi để tôi có thể tránh mất thông tin quan trọng.

**Acceptance Criteria:**

* Dữ liệu được sao lưu tự động định kỳ hàng ngày/tuần
* Có tùy chọn khôi phục dữ liệu từ bản sao lưu gần nhất
* Hiển thị trạng thái và lịch sử sao lưu cho người dùng
* Người dùng có thể cấu hình thời điểm và tần suất sao lưu

**Độ phức tạp (Story Point):** 18  
**Độ ưu tiên:** Trung bình  
**Có liên quan bảo mật:** Yes (bảo mật dữ liệu sao lưu)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup:**

* Toggle bật/tắt sao lưu tự động trong settings
* Dropdown chọn tần suất sao lưu (hàng ngày/tuần/tháng)
* Danh sách các bản sao lưu với thời gian tạo
* Button khôi phục dữ liệu

**Giao diện đầu vào/đầu ra:**

*Đầu vào:*

* userId: string (ID người dùng)
* backupFrequency: string (daily/weekly/monthly)
* autoBackup: boolean (Bật/tắt sao lưu tự động)
* backupId: string (ID bản sao lưu để khôi phục)

*Đầu ra:*

* Bản sao lưu được tạo theo lịch trình
* Danh sách các bản sao lưu có sẵn
* Trạng thái khôi phục dữ liệu

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn:** Tính năng sao lưu tự động giúp bảo vệ dữ liệu của bạn bằng cách tạo bản sao định kỳ và cho phép khôi phục khi cần.

**Bước sử dụng:**

*Để bật sao lưu tự động:*

1. Truy cập "Cài đặt" > "Sao lưu"
2. Bật tùy chọn "Tự động sao lưu"
3. Chọn lịch trình (hằng ngày, hằng tuần, hằng tháng)
4. Chọn thời gian thực hiện sao lưu
5. Lưu cài đặt

*Để khôi phục dữ liệu:*

1. Truy cập danh sách bản sao lưu
2. Chọn bản sao lưu muốn khôi phục
3. Nhấn nút "Khôi phục"
4. Xác nhận hành động

**Tooltip & hỗ trợ:**

* Tooltip: "Sao lưu tự động sẽ chạy vào thời gian bạn đã cài đặt"
* Hỗ trợ: "Khuyến nghị sao lưu hàng ngày để đảm bảo an toàn tối đa"

**Ghi chú kỹ thuật:**

* Dữ liệu sao lưu có thể lưu trong hệ thống file nội bộ hoặc cloud (AWS S3)
* Sử dụng định dạng nén để giảm dung lượng
* Lập trình với cronjob hoặc scheduler để tự động hóa
* Mã hóa dữ liệu sao lưu để bảo mật
* Framework đề xuất: Node.js với node-cron, AWS SDK cho cloud storage

## US20 – Tương tác với người đọc

### 1.1. Mô tả User Story

**Mã User Story**: US20

**Tên chức năng**: Tương tác với người đọc

**Mục đích**: Cho phép chủ blog quản lý bình luận và phản hồi từ người đọc để tăng cường tương tác trên blog.

**Câu chuyện người dùng**:

Là người dùng, tôi muốn quản lý các tương tác từ người đọc (bình luận, phản hồi) để cải thiện mức độ tương tác trên blog của tôi.

**Acceptance Criteria**:

* Người đọc có thể để lại bình luận công khai dưới bài viết.
* Chỉ các bình luận được phê duyệt mới hiển thị công khai.
* Chủ blog có thể trả lời từng bình luận.
* Hệ thống gửi thông báo khi có bình luận mới.

**Độ phức tạp (Story Point)**: 15

**Độ ưu tiên**: Cao

**Có liên quan bảo mật**: Yes (bảo vệ nội dung bình luận, chống spam)

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup**:

* **Comment box**: Ô nhập bình luận dưới mỗi bài viết.
* **Admin panel**: Danh sách bình luận chờ duyệt với các nút [Phê duyệt], [Xóa], [Trả lời].
* **Notification**: Thông báo hiển thị khi có bình luận mới (trên dashboard hoặc qua email/đẩy).
* **Comment thread**: Hiển thị phản hồi của chủ blog dưới bình luận chính.

**Giao diện đầu vào/đầu ra**:

**Đầu vào**:

* userId: string (ID người dùng, có thể là người đọc hoặc chủ blog)
* postId: string (ID bài viết)
* commentContent: string (Nội dung bình luận)
* approveStatus: boolean (Trạng thái phê duyệt bình luận)
* notificationPreference: string (email/push/none – tùy chọn thông báo)

**Đầu ra**:

* Bình luận được lưu và chờ phê duyệt.
* Bình luận được hiển thị công khai sau khi phê duyệt.
* Phản hồi của chủ blog hiển thị dưới bình luận chính.
* Thông báo được gửi khi có bình luận mới.

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn**:

Tính năng tương tác với người đọc cho phép quản lý bình luận, phê duyệt và trả lời bình luận, đồng thời nhận thông báo khi có bình luận mới để tăng cường tương tác trên blog.

**Bước sử dụng**:

**Để quản lý bình luận**:

1. Truy cập “Dashboard” > “Bình luận”.
2. Xem danh sách bình luận chờ duyệt.
3. Nhấn [Phê duyệt] để hiển thị công khai, [Xóa] để loại bỏ, hoặc [Trả lời] để phản hồi bình luận.

**Để cấu hình thông báo**:

1. Truy cập “Cài đặt” > “Thông báo”.
2. Bật tùy chọn nhận thông báo qua email hoặc thông báo đẩy khi có bình luận mới.
3. Lưu cài đặt.

**Hình ảnh minh họa**:

A black and grey rectangular object

AI-generated content may be incorrect.

Hình 37. Tương tác với người đọc

**Tooltip & hỗ trợ**:

* **Tooltip**: “Chỉ các bình luận được phê duyệt mới hiển thị công khai trên blog.”
* **Hỗ trợ**: “Khuyến nghị bật thông báo đẩy để phản hồi bình luận nhanh chóng.”

**Ghi chú kỹ thuật**:

* Lưu thông tin bình luận (thời gian gửi, người gửi, nội dung) trong cơ sở dữ liệu (e.g., MongoDB).
* Sử dụng WebSocket hoặc push notification (e.g., Firebase) để thông báo thời gian thực.
* Triển khai cơ chế chống spam (e.g., CAPTCHA hoặc giới hạn tần suất bình luận).
* Framework đề xuất: Node.js với Socket.IO cho thông báo thời gian thực, MongoDB cho lưu trữ dữ liệu.

## US21 – Tùy chỉnh giao diện blog

### 1.1. Mô tả User Story

**Mã User Story**: US21

**Tên chức năng**: Tùy chỉnh giao diện blog

**Mục đích**: Cho phép người dùng tùy chỉnh giao diện blog để phù hợp với thương hiệu và phong cách cá nhân.

**Câu chuyện người dùng**:

Là người dùng, tôi muốn tùy chỉnh giao diện blog của mình để nó phản ánh phong cách cá nhân và thương hiệu của tôi.

**Acceptance Criteria**:

* Người dùng có thể thay đổi màu sắc và font chữ của giao diện.
* Người dùng có thể sắp xếp lại bố cục các thành phần trên blog.
* Các thay đổi giao diện được lưu lại và áp dụng sau khi lưu.
* Người dùng có thể xem trước giao diện trước khi lưu.

**Độ phức tạp (Story Point)**: 13

**Độ ưu tiên**: Trung bình

**Có liên quan bảo mật**: No

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup**:

* **Color picker**: Công cụ chọn màu sắc cho nền, văn bản, v.v.
* **Font selector**: Dropdown để chọn font chữ (e.g., Arial, Roboto).
* **Drag & drop layout editor**: Giao diện kéo thả để sắp xếp bố cục.
* **Button**: [Xem trước], [Lưu].

**Giao diện đầu vào/đầu ra**:

**Đầu vào**:

* userId: string (ID người dùng)
* themeConfig: object (Cấu hình giao diện: màu sắc, font chữ, bố cục)
* preview: boolean (Xem trước hay lưu)

**Đầu ra**:

* Giao diện blog được cập nhật theo cấu hình.
* Xem trước giao diện trong modal hoặc iframe.
* Thông báo trạng thái lưu thành công.

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn**:

Tính năng tùy chỉnh giao diện cho phép người dùng thay đổi màu sắc, font chữ và bố cục blog để phù hợp với phong cách cá nhân.

**Bước sử dụng**:

**Để tùy chỉnh giao diện**:

1. Truy cập “Cài đặt” > “Tùy chỉnh giao diện”.
2. Chọn màu sắc, font chữ và bố cục mong muốn thông qua color picker, font selector và drag & drop editor.
3. Nhấn nút “Xem trước” để kiểm tra giao diện.
4. Nhấn nút “Lưu” để áp dụng thay đổi.

**Tooltip & hỗ trợ**:

* **Tooltip**: “Sử dụng ‘Xem trước’ để kiểm tra giao diện trước khi lưu.”
* **Hỗ trợ**: “Khuyến nghị chọn font chữ dễ đọc và màu sắc hài hòa để tối ưu trải nghiệm người dùng.”

**Ghi chú kỹ thuật**:

* Sử dụng CSS variables để áp dụng thay đổi giao diện động.
* Lưu cấu hình giao diện trong cơ sở dữ liệu (e.g., MongoDB).
* Sử dụng iframe hoặc modal cho chế độ xem trước.
* Framework đề xuất: React.js với styled-components cho giao diện, Node.js cho backend.

## US22 – Đề xuất nội dung liên quan

### 1.1. Mô tả User Story

**Mã User Story**: US22

**Tên chức năng**: Đề xuất nội dung liên quan

**Mục đích**: Tăng thời gian đọc và tương tác của người dùng bằng cách đề xuất các bài viết liên quan.

**Câu chuyện người dùng**:

Là người dùng, tôi muốn hệ thống đề xuất các bài viết liên quan để tăng thời gian đọc và mức độ tương tác của tôi.

**Acceptance Criteria**:

* Hiển thị ít nhất 3 bài viết liên quan dưới mỗi bài đăng.
* Bài viết liên quan được đề xuất dựa trên tag hoặc danh mục.
* Người dùng có thể nhấp vào bài viết đề xuất để chuyển đến nội dung.
* Hiển thị thông báo khi không có bài viết liên quan.

**Độ phức tạp (Story Point)**: 10

**Độ ưu tiên**: Cao

**Có liên quan bảo mật**: No

### 1.2. Thiết kế chức năng theo Story

**Wireframe/Mockup**:

* **Section**: “Bài viết liên quan” (hiển thị ở cuối bài viết hoặc sidebar).
* **Card**: Hiển thị thumbnail, tiêu đề bài viết và link điều hướng.
* **Thông báo**: Hiển thị “Không có bài viết liên quan” khi không có kết quả.

**Giao diện đầu vào/đầu ra**:

**Đầu vào**:

* postId: string (ID bài viết hiện tại)
* tags: array (Danh sách tag của bài viết)
* category: string (Danh mục của bài viết)

**Đầu ra**:

* Danh sách bài viết liên quan (ít nhất 3 bài).
* Thông báo khi không có bài viết liên quan.
* Điều hướng chính xác khi nhấp vào bài viết đề xuất.

### 1.3. Tài liệu sử dụng (User Guide)

**Mô tả ngắn**:

Tính năng đề xuất nội dung liên quan tự động hiển thị các bài viết có cùng tag hoặc danh mục để tăng tương tác của người dùng.

**Bước sử dụng**:

* Không cần cấu hình thủ công.
* Khi đọc một bài viết, hệ thống tự động hiển thị danh sách “Bài viết liên quan” ở cuối trang hoặc sidebar.
* Nhấp vào bài viết đề xuất để đọc nội dung.

**Tooltip & hỗ trợ**:

* **Tooltip**: “Bài viết liên quan được đề xuất dựa trên danh mục và từ khóa.”
* **Hỗ trợ**: “Khuyến nghị sử dụng tag và danh mục cụ thể để có đề xuất chính xác hơn.”

**Ghi chú kỹ thuật**:

* Sử dụng ElasticSearch hoặc full-text search để tìm kiếm bài viết liên quan dựa trên tag và danh mục.
* Cache kết quả đề xuất để tối ưu hiệu suất.
* Ưu tiên hiển thị các bài viết có lượt xem và tương tác cao.
* Framework đề xuất: Node.js với ElasticSearch, Redis cho caching.

# CHƯƠNG 5. TÀI LIỆU KIỂM THỬ

## I. Sprint 1

### 1. Kế hoạch kiểm thử

**Mục tiêu:** Kiểm thử các chức năng cốt lõi được phát triển trong Sprint 1, bao gồm:

* Đăng ký tài khoản người dùng
* Đăng nhập vào hệ thống với thông tin xác thực
* Quản lý phiên làm việc bảo mật
* Thiết lập bảo mật hệ thống (mã hóa dữ liệu, cấu hình API)
* Thiết lập nền tảng giao diện responsive cơ bản

**Phạm vi kiểm thử:**

* Chức năng phía người dùng (frontend)
* API xác thực và quản lý phiên làm việc
* API bảo mật
* Kiểm thử đơn lẻ (Unit test) + Kiểm thử chức năng (Functional test)

**Chiến lược:**

* Viết test case theo từng User Story
* Sử dụng công cụ kiểm thử thủ công (manual test)
* Ghi nhận lỗi bằng Jira

**Nhân sự:**

* Tester: Quốc
* Developer: Quốc, Trường, Tuấn, Bảo
* Scrum Master: Quốc

**Công cụ:**

* Trình duyệt: Chrome (phiên bản mới nhất), Firefox (phiên bản mới nhất)
* Hệ điều hành: Windows 11
* Thiết bị di động: Android (phiên bản mới nhất)
* Công cụ ghi lỗi: Jira

### 2. Test Cases

Bảng 17. Bảng các test case của sprint 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Test Case** | **Bước kiểm thử** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC01.01 | US01 - Đăng ký | Kiểm tra đăng ký tài khoản thành công với thông tin hợp lệ | 1. Truy cập trang đăng ký.  2. Nhập email hợp lệ (ví dụ: [test@example.com](mailto:test@example.com)).  3. Nhập mật khẩu hợp lệ (ví dụ: Password123!).  4. Nhấn nút "Đăng ký". | Hệ thống hiển thị thông báo đăng ký thành công và chuyển hướng đến trang đăng nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.02 | US01 - Đăng ký | Kiểm tra đăng ký với email đã tồn tại | 1. Truy cập trang đăng ký.  2. Nhập email đã được đăng ký trước đó.  3. Nhập mật khẩu hợp lệ.  4. Nhấn nút "Đăng ký". | Hệ thống hiển thị thông báo lỗi "Email đã tồn tại". | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.03 | US01 - Đăng ký | Kiểm tra đăng ký với email sai định dạng | 1. Truy cập trang đăng ký.  2. Nhập email sai định dạng (ví dụ: abc@).  3. Nhập mật khẩu hợp lệ.  4. Nhấn nút "Đăng ký". | Hệ thống hiển thị thông báo lỗi "Email không hợp lệ". | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.04 | US01 - Đăng nhập | Kiểm tra đăng nhập thành công với thông tin hợp lệ | 1. Truy cập trang đăng nhập.  2. Nhập email và mật khẩu của tài khoản đã đăng ký thành công.  3. Nhấn nút "Đăng nhập". | Hệ thống chuyển hướng đến trang dashboard/trang chính của AutoBlog. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.05 | US01 - Đăng nhập | Kiểm tra đăng nhập với mật khẩu sai | 1. Truy cập trang đăng nhập.  2. Nhập email hợp lệ.  3. Nhập mật khẩu không đúng.  4. Nhấn nút "Đăng nhập". | Hệ thống hiển thị thông báo lỗi "Tên đăng nhập hoặc mật khẩu không đúng". | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.06 | US01 - Đăng nhập | Kiểm tra đăng nhập với tài khoản chưa kích hoạt (nếu có chức năng kích hoạt) | 1. Truy cập trang đăng nhập.  2. Nhập email và mật khẩu của tài khoản chưa được kích hoạt.  3. Nhấn nút "Đăng nhập". | Hệ thống hiển thị thông báo lỗi "Tài khoản chưa được kích hoạt". | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.07 | US18 - Bảo mật | Kiểm tra mã hóa mật khẩu khi đăng ký/đăng nhập (kiểm tra trong database hoặc network request) | 1. Đăng ký/Đăng nhập một tài khoản.  2. Kiểm tra dữ liệu mật khẩu được lưu trữ trong database (nếu có quyền truy cập).  3. Kiểm tra payload của network request khi đăng nhập/đăng ký. | Mật khẩu không được lưu trữ dưới dạng văn bản thuần túy (plain text) trong database. Mật khẩu được mã hóa/hash trong network request. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.08 | US18 - Bảo mật | Kiểm tra quản lý phiên làm việc sau khi đăng xuất | 1. Đăng nhập thành công.  2. Đăng xuất khỏi hệ thống.  3. Thử truy cập lại một trang yêu cầu đăng nhập bằng cách sử dụng nút back của trình duyệt hoặc nhập URL trực tiếp. | Người dùng được yêu cầu đăng nhập lại, không thể truy cập các trang bảo mật. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.09 | US15.1 - Giao diện | Kiểm tra hiển thị trang đăng ký/đăng nhập trên thiết bị di động (responsive) | 1. Truy cập trang đăng ký/đăng nhập trên trình duyệt di động hoặc sử dụng chế độ giả lập thiết bị trong công cụ phát triển trình duyệt.  2. Thay đổi kích thước màn hình. | Giao diện hiển thị tốt, không bị tràn, các thành phần co giãn phù hợp. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.10 | US15.1 - Giao diện | Kiểm tra hiển thị trang đăng ký/đăng nhập trên máy tính bảng (responsive) | 1. Truy cập trang đăng ký/đăng nhập trên trình duyệt máy tính bảng hoặc sử dụng chế độ giả lập thiết bị trong công cụ phát triển trình duyệt.  2. Thay đổi kích thước màn hình. | Giao diện hiển thị tốt, không bị tràn, các thành phần co giãn phù hợp. | Như mong đợi | ✅ Đạt yêu cầu |
| TC01.11 | US18 - Bảo mật | Kiểm tra khả năng chặn các cuộc tấn công CSRF (ví dụ: gửi request mà không có token CSRF) | 1. Cố gắng gửi một POST request đến API đăng nhập/đăng ký mà không có hoặc với token CSRF không hợp lệ. | Hệ thống từ chối request và trả về lỗi bảo mật (ví dụ: 403 Forbidden). | Không cảnh báo | ❌ Không đạt yêu cầu |

### 3. Báo cáo bug

Bảng 18. Bảng các bug của sprint 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mã Bug** | **Mô tả** | **Độ nghiêm trọng** | **Trạng thái** | **Người xử lý** | **Ghi chú** |
| BUG-001-S1 | Hệ thống chưa chặn được các cuộc tấn công CSRF khi đăng nhập/đăng ký. | High | Open | Quốc | Cần triển khai cơ chế bảo vệ CSRF để tăng cường bảo mật. |

### 4. Báo cáo tổng quan kết quả kiểm thử

**Tổng số test case:** 11

**Đạt yêu cầu:** 10

**Không đạt yêu cầu:** 1

**Bug nghiêm trọng:** 1

**Bug trung bình:** 0

**Kết luận:** Sprint 1 đã hoàn thành các mục tiêu cơ bản về đăng ký, đăng nhập và nền tảng giao diện responsive. Chức năng bảo mật cơ bản đã được triển khai, tuy nhiên cần khắc phục lỗi liên quan đến bảo vệ CSRF để đảm bảo an toàn cho hệ thống. Nhóm phát triển cần ưu tiên khắc phục BUG-001-S1 trong thời gian sớm nhất. Hệ thống tạm ổn định, tiếp tục cải tiến ở Sprint 2.

## II. Sprint 2

### 1. Kế hoạch kiểm thử

**Mục tiêu:** Kiểm thử các chức năng quản lý nội dung cơ bản được phát triển trong Sprint 2, bao gồm:

* Tạo bài viết mới với tiêu đề, nội dung và hình ảnh
* Chỉnh sửa bài viết hiện có
* Xóa bài viết không cần thiết
* Quản lý hình ảnh và tệp đính kèm (tải lên, xóa, tích hợp vào bài viết)
* Phân loại bài viết theo danh mục (tạo, chỉnh sửa, xóa danh mục, gán bài viết vào danh mục)
* Đảm bảo giao diện hiển thị responsive cho tất cả các chức năng quản lý nội dung

**Phạm vi kiểm thử:**

* Chức năng phía người dùng (frontend) cho việc tạo, chỉnh sửa, xóa bài viết, quản lý hình ảnh và danh mục
* API cho việc tạo, cập nhật, xóa bài viết, quản lý media và CRUD danh mục
* Kiểm thử đơn lẻ (Unit test) + Kiểm thử chức năng (Functional test)

**Chiến lược:**

* Viết test case theo từng User Story
* Sử dụng công cụ kiểm thử thủ công (manual test)
* Ghi nhận lỗi bằng Jira

**Nhân sự:**

* Tester: Quốc
* Developer: Quốc, Trường, Tuấn, Bảo
* Scrum Master: Quốc

**Công cụ:**

* Trình duyệt: Chrome (phiên bản mới nhất), Firefox (phiên bản mới nhất)
* Hệ điều hành: Windows 11
* Thiết bị di động: Android (phiên bản mới nhất)
* Công cụ ghi lỗi: Jira

### 2. Test Cases

Bảng 19. Bảng các test case của sprint 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Test Case** | **Bước kiểm thử** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC02.01 | US02 - Tạo bài viết mới | Kiểm tra tạo bài viết mới thành công với đầy đủ thông tin | 1. Đăng nhập vào hệ thống.  2. Truy cập trang "Tạo bài viết mới".  3. Nhập tiêu đề, nội dung, và tải lên một hình ảnh.  4. Chọn một danh mục (nếu có).  5. Nhấn nút "Lưu bài viết". | Bài viết được lưu thành công vào cơ sở dữ liệu và hiển thị trong danh sách bài viết. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.02 | US02 - Tạo bài viết mới | Kiểm tra tạo bài viết không có tiêu đề | 1. Đăng nhập vào hệ thống.  2. Truy cập trang "Tạo bài viết mới".  3. Bỏ trống tiêu đề, nhập nội dung, và tải lên hình ảnh (tùy chọn).  4. Nhấn nút "Lưu bài viết". | Hệ thống hiển thị cảnh báo yêu cầu nhập tiêu đề. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.03 | US02 - Tạo bài viết mới | Kiểm tra tạo bài viết với nội dung trống | 1. Đăng nhập vào hệ thống.   2. Truy cập trang "Tạo bài viết mới".  3. Nhập tiêu đề, bỏ trống nội dung, và tải lên hình ảnh (tùy chọn).  4. Nhấn nút "Lưu bài viết". | Hệ thống hiển thị cảnh báo yêu cầu nhập nội dung (tùy thuộc quy định). | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.04 | US04 - Chỉnh sửa bài viết | Kiểm tra chỉnh sửa nội dung bài viết thành công | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết hiện có từ danh sách.  3. Nhấn nút "Chỉnh sửa".  4. Thay đổi tiêu đề và/hoặc nội dung của bài viết.  5. Nhấn nút "Lưu". | Bài viết được cập nhật thành công, phiên bản mới thay thế phiên bản cũ. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.05 | US04 - Chỉnh sửa bài viết | Kiểm tra chỉnh sửa bài viết không thay đổi nội dung | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết hiện có.  3. Nhấn nút "Chỉnh sửa".  4. Không thay đổi bất kỳ nội dung nào.  5. Nhấn nút "Lưu". | Hệ thống lưu thành công mà không báo lỗi, hoặc thông báo không có gì thay đổi. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.06 | US05 - Xóa bài viết | Kiểm tra xóa bài viết thành công | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết từ danh sách.  3. Nhấn nút "Xóa".  4. Xác nhận hành động xóa khi có hộp thoại. | Bài viết bị xóa khỏi hệ thống và không còn hiển thị trong danh sách. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.07 | US05 - Xóa bài viết | Kiểm tra hủy bỏ thao tác xóa bài viết | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết từ danh sách.  3. Nhấn nút "Xóa". 4. Nhấn "Hủy" hoặc "Không" trên hộp thoại xác nhận. | Bài viết vẫn còn trong danh sách và không bị xóa. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.08 | US09 - Quản lý hình ảnh | Kiểm tra tải lên hình ảnh thành công | 1. Đăng nhập vào hệ thống.  2. Truy cập chức năng quản lý hình ảnh hoặc trong trình soạn thảo bài viết.  3. Chọn một tệp hình ảnh hợp lệ (ví dụ: .jpg, .png).  4. Nhấn nút "Tải lên". | Hình ảnh được tải lên thành công và hiển thị trong thư viện media. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.09 | US09 - Quản lý hình ảnh | Kiểm tra tải lên tệp không phải hình ảnh (ví dụ: .exe, .zip) | 1. Đăng nhập vào hệ thống.  2. Truy cập chức năng quản lý hình ảnh.  3. Cố gắng tải lên một tệp không được hỗ trợ. | Hệ thống hiển thị thông báo lỗi về định dạng tệp không hợp lệ. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.10 | US09 - Quản lý hình ảnh | Kiểm tra tích hợp hình ảnh vào bài viết từ thư viện | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết.  3. Mở thư viện media và chọn một hình ảnh đã tải lên.  4. Chèn hình ảnh vào nội dung bài viết. | Hình ảnh được chèn vào nội dung bài viết và hiển thị chính xác. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.11 | US06 - Phân loại bài viết theo danh mục | Kiểm tra tạo danh mục mới thành công | 1. Đăng nhập vào hệ thống.  2. Truy cập trang quản lý danh mục.  3. Nhập tên danh mục mới.  4. Nhấn nút "Thêm danh mục". | Danh mục mới được tạo và hiển thị trong danh sách. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.12 | US06 - Phân loại bài viết theo danh mục | Kiểm tra chỉnh sửa tên danh mục | 1. Đăng nhập vào hệ thống.  2. Truy cập trang quản lý danh mục.  3. Chọn một danh mục hiện có.  4. Chỉnh sửa tên danh mục.  5. Nhấn nút "Lưu". | Tên danh mục được cập nhật thành công. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.13 | US06 - Phân loại bài viết theo danh mục | Kiểm tra xóa danh mục | 1. Đăng nhập vào hệ thống.  2. Truy cập trang quản lý danh mục.  3. Chọn một danh mục.  4. Nhấn nút "Xóa".  5. Xác nhận xóa. | Danh mục bị xóa khỏi hệ thống. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.14 | US06 - Phân loại bài viết theo danh mục | Kiểm tra gán bài viết vào danh mục | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết.  3. Trong form bài viết, chọn một hoặc nhiều danh mục từ danh sách.  4. Lưu bài viết. | Bài viết được gán vào các danh mục đã chọn và hiển thị chính xác khi lọc theo danh mục. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.15 | US15.2 - Responsive cho quản lý nội dung | Kiểm tra giao diện tạo/chỉnh sửa bài viết trên mobile | 1. Truy cập trang tạo/chỉnh sửa bài viết trên thiết bị di động hoặc giả lập mobile.  2. Thao tác nhập liệu, tải ảnh. | Trình soạn thảo hiển thị tốt, các nút chức năng không bị che khuất, form responsive. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.16 | US15.2 - Responsive cho quản lý nội dung | Kiểm tra giao diện quản lý danh mục trên tablet | 1. Truy cập trang quản lý danh mục trên máy tính bảng hoặc giả lập tablet.  2. Thao tác tạo, chỉnh sửa, xóa danh mục. | Bảng danh sách và các nút thao tác hiển thị rõ ràng, dễ sử dụng. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.17 | US15.2 - Responsive cho quản lý nội dung | Kiểm tra giao diện xác nhận xóa bài viết trên các thiết bị | 1. Truy cập trang quản lý bài viết.  2. Nhấn nút xóa một bài viết.  3. Xem hộp thoại xác nhận trên các kích thước màn hình khác nhau (desktop, tablet, mobile). | Hộp thoại xác nhận hiển thị căn giữa, nội dung rõ ràng, các nút bấm dễ thao tác. | Như mong đợi | ✅ Đạt yêu cầu |
| TC02.18 | US02 - Tạo bài viết mới | Kiểm tra tạo bài viết với nội dung vượt quá giới hạn ký tự (nếu có giới hạn) | 1. Đăng nhập vào hệ thống.  2. Truy cập trang "Tạo bài viết mới".  3. Nhập tiêu đề và nội dung vượt quá giới hạn cho phép (ví dụ: 10.000 ký tự).  4. Nhấn nút "Lưu bài viết". | Hệ thống hiển thị cảnh báo lỗi về giới hạn ký tự và không cho phép lưu. | Không cảnh báo | ❌ Không đạt yêu cầu |

### 3. Báo cáo bug

Bảng 20. Bảng các bug của sprint 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mã Bug** | **Mô tả** | **Độ nghiêm trọng** | **Trạng thái** | **Người xử lý** | **Ghi chú** |
| BUG-001-S2 | Hệ thống không hiển thị cảnh báo khi người dùng nhập nội dung bài viết vượt quá giới hạn ký tự cho phép. | Medium | Open | Trường | Cần thêm validation cho trường nội dung bài viết để giới hạn ký tự và cảnh báo người dùng. |

### 4. Báo cáo tổng quan kết quả kiểm thử

**Tổng số test case:** 18

**Đạt yêu cầu:** 17

**Không đạt yêu cầu:** 1

**Bug nghiêm trọng:** 0

**Bug trung bình:** 1

**Kết luận:** Sprint 2 đã hoàn thành xuất sắc các chức năng CRUD cơ bản cho bài viết, quản lý hình ảnh và phân loại danh mục. Giao diện responsive cũng hoạt động tốt trên các thiết bị khác nhau. Tuy nhiên, một lỗi mức độ trung bình đã được phát hiện liên quan đến việc không kiểm tra giới hạn ký tự cho nội dung bài viết. Nhóm phát triển cần xử lý BUG-001-S2 để cải thiện độ bền và tính ổn định của hệ thống. Nhìn chung, hệ thống đang tiến triển tốt và sẵn sàng cho Sprint 3.

## III. Sprint 3

### 1. Kế hoạch kiểm thử

**Mục tiêu:** Kiểm thử các chức năng nâng cao trải nghiệm người dùng và tích hợp AI được phát triển trong Sprint 3, bao gồm:

* Xem trước bài viết trước khi đăng
* Tìm kiếm bài viết theo từ khóa, tiêu đề hoặc nội dung
* Lọc và sắp xếp bài viết theo danh mục hoặc ngày tháng
* Tự động tạo nội dung dựa trên từ khóa và chủ đề do người dùng cung cấp
* Đảm bảo giao diện tìm kiếm và lọc thân thiện trên mọi thiết bị

**Phạm vi kiểm thử:**

* Chức năng phía người dùng (frontend) cho xem trước bài viết, tìm kiếm, lọc, sắp xếp, và tạo nội dung AI
* API cho việc tìm kiếm, lọc, sắp xếp bài viết và tích hợp API AI
* Kiểm thử đơn lẻ (Unit test) + Kiểm thử chức năng (Functional test)

**Chiến lược:**

* Viết test case theo từng User Story
* Sử dụng công cụ kiểm thử thủ công (manual test)
* Ghi nhận lỗi bằng Jira

**Nhân sự:**

* Tester: Quốc
* Developer: Quốc, Trường, Tuấn, Bảo
* Scrum Master: Quốc

**Công cụ:**

* Trình duyệt: Chrome (phiên bản mới nhất), Firefox (phiên bản mới nhất)
* Hệ điều hành: Windows 11
* Thiết bị di động: Android (phiên bản mới nhất)
* Công cụ ghi lỗi: Jira

### 2. Test Cases

Bảng 21. Bảng các test case của sprint 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Test Case** | **Bước kiểm thử** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC03.01 | US16 - Chế độ xem trước bài viết | Kiểm tra chế độ xem trước hiển thị chính xác nội dung và định dạng bài viết | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết với nội dung (text, hình ảnh, định dạng).  3. Nhấn nút "Xem trước". | Bài viết hiển thị với định dạng và nội dung chính xác như khi được đăng tải. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.02 | US16 - Chế độ xem trước bài viết | Kiểm tra quay lại chỉnh sửa từ chế độ xem trước | 1. Từ chế độ xem trước bài viết.  2. Nhấn nút "Quay lại chỉnh sửa" (hoặc tương tự). | Hệ thống quay trở lại giao diện chỉnh sửa bài viết với nội dung đã nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.03 | US10 - Tìm kiếm bài viết | Kiểm tra tìm kiếm bài viết theo từ khóa trong tiêu đề | 1. Đăng nhập vào hệ thống.  2. Nhập một từ khóa có trong tiêu đề của một bài viết hiện có vào ô tìm kiếm.  3. Nhấn Enter hoặc nút "Tìm kiếm". | Danh sách kết quả hiển thị các bài viết có tiêu đề chứa từ khóa đã nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.04 | US10 - Tìm kiếm bài viết | Kiểm tra tìm kiếm bài viết theo từ khóa trong nội dung | 1. Đăng nhập vào hệ thống.  2. Nhập một từ khóa có trong nội dung của một bài viết hiện có vào ô tìm kiếm.  3. Nhấn Enter hoặc nút "Tìm kiếm". | Danh sách kết quả hiển thị các bài viết có nội dung chứa từ khóa đã nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.05 | US10 - Tìm kiếm bài viết | Kiểm tra tìm kiếm không có kết quả | 1. Đăng nhập vào hệ thống.  2. Nhập một từ khóa không có trong bất kỳ bài viết nào.  3. Nhấn Enter hoặc nút "Tìm kiếm". | Hệ thống hiển thị thông báo "Không tìm thấy kết quả" hoặc danh sách trống. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.06 | US07 - Lọc và sắp xếp bài viết | Kiểm tra lọc bài viết theo danh mục | 1. Đăng nhập vào hệ thống.  2. Chọn một danh mục từ bộ lọc. | Danh sách bài viết chỉ hiển thị các bài viết thuộc danh mục đã chọn. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.07 | US07 - Lọc và sắp xếp bài viết | Kiểm tra sắp xếp bài viết theo ngày đăng (mới nhất/cũ nhất) | 1. Đăng nhập vào hệ thống.  2. Chọn tùy chọn sắp xếp "Ngày đăng mới nhất" (hoặc "Cũ nhất"). | Danh sách bài viết được sắp xếp chính xác theo thứ tự ngày đăng giảm dần (hoặc tăng dần). | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.08 | US07 - Lọc và sắp xếp bài viết | Kiểm tra kết hợp lọc và sắp xếp | 1. Đăng nhập vào hệ thống.  2. Chọn một danh mục từ bộ lọc.  3. Chọn tùy chọn sắp xếp (ví dụ: "Ngày đăng cũ nhất"). | Danh sách bài viết hiển thị các bài viết thuộc danh mục đã chọn và được sắp xếp chính xác. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.09 | US03 - Tự động tạo nội dung AI | Kiểm tra tạo nội dung AI dựa trên từ khóa và hiển thị trong chatbox | 1. Đăng nhập vào hệ thống.  2. Truy cập chức năng tạo nội dung AI (khu vực chatbox).  3. Nhập yêu cầu/câu hỏi chứa từ khóa (ví dụ: "Viết về lợi ích của AI") vào chatbox.  4. Gửi yêu cầu. | AI chatbox trả về một đoạn nội dung có liên quan đến từ khóa đã nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.10 | US03 - Tự động tạo nội dung AI | Kiểm tra tạo nội dung AI dựa trên chủ đề và hiển thị trong chatbox | 1. Đăng nhập vào hệ thống.  2. Truy cập chức năng tạo nội dung AI (khu vực chatbox).  3. Nhập yêu cầu/câu hỏi chứa chủ đề (ví dụ: "Tạo bài viết hướng dẫn chăm sóc cây cảnh") vào chatbox.  4. Gửi yêu cầu. | AI chatbox trả về một đoạn nội dung có liên quan đến chủ đề đã nhập. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.11 | US03 - Tự động tạo nội dung AI | Kiểm tra áp dụng nội dung từ chatbox vào trình soạn thảo và chỉnh sửa trước khi lưu | 1. Sau khi nội dung AI được tạo và hiển thị trong chatbox.  2. Nhấn nút "Áp dụng vào trường nội dung" (hoặc tên tương tự) bên cạnh nội dung AI.  3. Kiểm tra nội dung đã được điền vào trình soạn thảo bài viết.  4. Chỉnh sửa nội dung đã điền trong trình soạn thảo.  5. Nhấn nút "Lưu bài viết". | Nội dung AI được điền chính xác vào trình soạn thảo, có thể chỉnh sửa và lưu lại thành công. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.12 | US03 - Tự động tạo nội dung AI | Kiểm tra phản hồi của AI với yêu cầu không rõ ràng/không phổ biến | 1. Đăng nhập vào hệ thống.  2. Truy cập chức năng tạo nội dung AI (khu vực chatbox).  3. Nhập yêu cầu/câu hỏi rất chung chung, không rõ ràng hoặc không phổ biến (ví dụ: "Hãy viết về zzzyyy", "Cho tôi vài ý tưởng về sự vô định hình của ý tưởng").  4. Gửi yêu cầu. | AI chatbox trả về thông báo rõ ràng rằng không thể tạo nội dung, hoặc yêu cầu cung cấp thêm thông tin, hoặc nội dung không liên quan (nhưng vẫn là phản hồi có ý nghĩa, không phải lỗi hệ thống). | AI trả về nội dung không liên quan hoặc thông báo lỗi chung chung (ví dụ: "Có lỗi xảy ra") | ❌ Không đạt yêu cầu |
| TC03.13 | US15.3 - Giao diện tìm kiếm và lọc responsive | Kiểm tra giao diện tìm kiếm trên mobile | 1. Truy cập trang danh sách bài viết trên thiết bị di động hoặc giả lập mobile.  2. Thao tác tìm kiếm. | Ô tìm kiếm, nút tìm kiếm và hiển thị kết quả tìm kiếm hiển thị tốt, dễ sử dụng. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.14 | US15.3 - Giao diện tìm kiếm và lọc responsive | Kiểm tra giao diện bộ lọc và sắp xếp trên mobile | 1. Truy cập trang danh sách bài viết trên thiết bị di động hoặc giả lập mobile.  2. Thao tác với các tùy chọn lọc và sắp xếp. | Các bộ lọc và tùy chọn sắp xếp hiển thị rõ ràng, dễ thao tác trên màn hình nhỏ. | Như mong đợi | ✅ Đạt yêu cầu |
| TC03.15 | US15.3 - Giao diện tìm kiếm và lọc responsive | Kiểm tra hiển thị kết quả tìm kiếm trên tablet | 1. Thực hiện tìm kiếm bài viết trên máy tính bảng hoặc giả lập tablet. | Kết quả tìm kiếm hiển thị dạng lưới hoặc danh sách phù hợp với kích thước màn hình tablet. | Như mong đợi | ✅ Đạt yêu cầu |

### 3. Báo cáo bug

Bảng 22. Bảng các bug của sprint 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mã Bug** | **Mô tả** | **Độ nghiêm trọng** | **Trạng thái** | **Người xử lý** | **Ghi chú** |
| BUG-001-S3 | Chức năng tạo nội dung AI trả về nội dung không liên quan hoặc thông báo lỗi khi nhập từ khóa/chủ đề quá chung chung hoặc không phổ biến. | Medium | Open | Bảo | Cần cải thiện khả năng xử lý các từ khóa/chủ đề ít thông tin của AI hoặc cung cấp phản hồi rõ ràng hơn cho người dùng. |

### 4. Báo cáo tổng quan kết quả kiểm thử

**Tổng số test case:** 15

**Đạt yêu cầu:** 14

**Không đạt yêu cầu:** 1

**Bug nghiêm trọng:** 0

**Bug trung bình:** 1

**Kết luận:** Sprint 3 đã thành công trong việc triển khai các chức năng nâng cao trải nghiệm người dùng như xem trước bài viết, tìm kiếm và lọc. Đặc biệt, việc tích hợp chức năng tạo nội dung AI là một bước tiến quan trọng. Tuy nhiên, một lỗi mức độ trung bình đã được phát hiện liên quan đến chất lượng phản hồi của AI khi đối mặt với các từ khóa/chủ đề ít thông tin. Nhóm phát triển cần xem xét cải thiện thuật toán AI hoặc xử lý các trường hợp ngoại lệ này để nâng cao trải nghiệm người dùng. Nhìn chung, sprint này đã đạt được các mục tiêu đề ra và sẵn sàng cho Sprint 4.

### IV. Sprint 4

Vì lí do thời gian nên nhóm không thực hiện sprint 4, tuy nhiên vẫn sẽ trình bày kế hoạch kiểm thử cùng các test case

### 1. Kết quả kiểm thử

**Mục tiêu:** Kiểm thử các chức năng tối ưu hóa SEO, tự động hóa quy trình đăng bài và sao lưu dữ liệu được phát triển trong Sprint 4, bao gồm:

* Tự động tối ưu hóa nội dung cho SEO (phân tích từ khóa, đề xuất cải thiện, tối ưu tiêu đề, meta description)
* Lên lịch đăng bài tự động
* Nhận thông báo khi bài viết được đăng thành công
* Nhập/xuất dữ liệu bài viết sang các định dạng phổ biến
* Đảm bảo giao diện lên lịch và SEO thân thiện trên mọi thiết bị

**Phạm vi kiểm thử:**

* Chức năng phía người dùng (frontend) cho tối ưu SEO, lên lịch đăng bài, thông báo, nhập/xuất dữ liệu
* API cho việc tối ưu SEO, quản lý lịch đăng bài, gửi thông báo, xử lý nhập/xuất dữ liệu
* Kiểm thử đơn lẻ (Unit test) + Kiểm thử chức năng (Functional test)

**Chiến lược:**

* Viết test case theo từng User Story
* Sử dụng công cụ kiểm thử thủ công (manual test)
* Ghi nhận lỗi bằng Jira

**Nhân sự:**

* Tester: Quốc
* Developer: Quốc, Trường, Tuấn, Bảo
* Scrum Master: Quốc

**Công cụ:**

* Trình duyệt: Chrome (phiên bản mới nhất), Firefox (phiên bản mới nhất)
* Hệ điều hành: Windows 11
* Thiết bị di động: Android (phiên bản mới nhất)
* Công cụ ghi lỗi: Jira

### 2. Test Cases

Bảng 23. Bảng các test case của sprint 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Test Case** | **Bước kiểm thử** | **Kết quả mong đợi** |
| TC04.01 | US11 - Tối ưu hóa SEO tự động | Kiểm tra hệ thống đề xuất cải thiện SEO cho tiêu đề | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết.  3. Nhập tiêu đề và nội dung.  4. Kiểm tra phần đề xuất SEO. | Hệ thống đề xuất các cải tiến cho tiêu đề (ví dụ: độ dài, từ khóa). |
| TC04.02 | US11 - Tối ưu hóa SEO tự động | Kiểm tra hệ thống đề xuất cải thiện SEO cho meta description | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết.  3. Nhập nội dung.  4. Kiểm tra phần đề xuất SEO. | Hệ thống đề xuất các cải tiến cho meta description (ví dụ: độ dài, từ khóa). |
| TC04.03 | US11 - Tối ưu hóa SEO tự động | Kiểm tra phân tích từ khóa chính trong nội dung | 1. Đăng nhập vào hệ thống.  2. Tạo hoặc chỉnh sửa một bài viết với nội dung có chứa từ khóa lặp lại.  3. Kiểm tra phần phân tích SEO. | Hệ thống nhận diện và hiển thị các từ khóa chính trong nội dung. |
| TC04.04 | US08 - Lên lịch đăng bài tự động | Kiểm tra lên lịch đăng bài thành công vào một thời điểm trong tương lai | 1. Đăng nhập vào hệ thống.  2. Tạo một bài viết mới.  3. Chọn tùy chọn "Lên lịch đăng bài" và thiết lập ngày giờ trong tương lai.  4. Lưu bài viết. | Bài viết được lưu với trạng thái "Đã lên lịch". Khi đến thời điểm đã chọn, bài viết tự động được đăng. |
| TC04.05 | US08 - Lên lịch đăng bài tự động | Kiểm tra sửa đổi lịch đăng bài | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết đã được lên lịch.  3. Thay đổi ngày giờ đăng bài.  4. Lưu thay đổi. | Lịch đăng bài được cập nhật thành công. |
| TC04.06 | US08 - Lên lịch đăng bài tự động | Kiểm tra hủy bỏ lịch đăng bài | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết đã được lên lịch.  3. Hủy bỏ lịch đăng bài (chuyển về trạng thái nháp).  4. Lưu thay đổi. | Bài viết trở về trạng thái nháp và không tự động đăng. |
| TC04.07 | US12 - Thông báo khi đăng bài thành công | Kiểm tra nhận thông báo qua email khi bài viết đăng thành công | 1. Lên lịch một bài viết và đợi đến thời điểm đăng.  2. Kiểm tra hộp thư email của người dùng. | Người dùng nhận được email thông báo "Bài viết đã đăng thành công". |
| TC04.08 | US12 - Thông báo khi đăng bài thành công | Kiểm tra nhận thông báo trong ứng dụng khi bài viết đăng thành công | 1. Lên lịch một bài viết và đợi đến thời điểm đăng.  2. Kiểm tra giao diện thông báo trong ứng dụng (ví dụ: icon chuông, danh sách thông báo). | Người dùng nhận được thông báo trong ứng dụng "Bài viết [Tên bài viết] đã được đăng thành công". |
| TC04.09 | US17 - Nhập/xuất dữ liệu | Kiểm tra xuất bài viết ra định dạng PDF | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết.  3. Chọn tùy chọn "Xuất" và định dạng "PDF". | Tệp PDF của bài viết được tải về máy tính và hiển thị đúng nội dung, định dạng. |
| TC04.10 | US17 - Nhập/xuất dữ liệu | Kiểm tra xuất bài viết ra định dạng Word (.docx) | 1. Đăng nhập vào hệ thống.  2. Chọn một bài viết.  3. Chọn tùy chọn "Xuất" và định dạng "Word". | Tệp Word của bài viết được tải về máy tính và hiển thị đúng nội dung, định dạng. |
| TC04.11 | US17 - Nhập/xuất dữ liệu | Kiểm tra xuất danh sách bài viết ra định dạng CSV | 1. Đăng nhập vào hệ thống.  2. Truy cập danh sách bài viết.  3. Chọn tùy chọn "Xuất CSV". | Tệp CSV chứa danh sách các bài viết (tiêu đề, danh mục, ngày đăng, v.v.) được tải về. |
| TC04.12 | US17 - Nhập/xuất dữ liệu | Kiểm tra nhập dữ liệu bài viết từ tệp CSV (nếu có) | 1. Chuẩn bị một tệp CSV với dữ liệu bài viết hợp lệ theo cấu trúc mẫu.  2. Đăng nhập vào hệ thống.  3. Chọn tùy chọn "Nhập dữ liệu".  4. Tải lên tệp CSV. | Các bài viết từ tệp CSV được nhập thành công vào hệ thống và hiển thị trong danh sách bài viết. |
| TC04.13 | US15.4 - Giao diện lên lịch và SEO responsive | Kiểm tra giao diện lên lịch đăng bài trên mobile | 1. Truy cập trang lên lịch đăng bài trên thiết bị di động hoặc giả lập mobile.  2. Thao tác chọn ngày giờ. | Giao diện chọn ngày giờ hiển thị tốt, dễ thao tác trên màn hình nhỏ. |
| TC04.14 | US15.4 - Giao diện lên lịch và SEO responsive | Kiểm tra hiển thị đề xuất SEO trên tablet | 1. Truy cập trang chỉnh sửa bài viết trên máy tính bảng hoặc giả lập tablet.  2. Kiểm tra phần đề xuất SEO. | Phần đề xuất SEO hiển thị rõ ràng, không bị tràn hay mất thông tin trên màn hình tablet. |

# CHƯƠNG 6. TÀI LIỆU CHẤP NHẬN (UAT)

## I. Sprint 1: Nền tảng cơ sở

### 1. Mục đích

* UAT Sprint 1 nhằm xác nhận rằng nền tảng cơ bản của hệ thống AutoBlog (bao gồm chức năng đăng ký, đăng nhập, bảo mật cơ bản và giao diện responsive) đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng đã đề ra.
* Đây là bước nghiệm thu quan trọng nhằm đảm bảo hệ thống ổn định, bảo mật và sẵn sàng để phát triển các tính năng phức tạp hơn trong các Sprint tiếp theo.

### 2. Phạm vi Thử nghiệm Chấp nhận

Các chức năng chính được kiểm thử trong Sprint 1 bao gồm:

* US01 – Đăng ký/Đăng nhập:
* Đăng ký tài khoản mới với dữ liệu hợp lệ.
* Đăng nhập thành công với thông tin đúng.
* Kiểm tra các trường hợp lỗi: sai mật khẩu, tài khoản không tồn tại, email trùng lặp.
* Quản lý phiên làm việc.
* US18 – Bảo mật hệ thống:
* Mã hóa mật khẩu đúng chuẩn (ví dụ: bcrypt).
* Sử dụng HTTPS/SSL trong kết nối.
* Kiểm tra các lỗ hổng như SQL Injection, XSS trong các form xác thực.
* Bảo vệ API xác thực với token hợp lệ.
* US15.1 – Giao diện thân thiện đa thiết bị:
* Giao diện hiển thị đúng trên desktop, tablet, mobile.
* Không vỡ layout, dễ thao tác trên màn hình nhỏ.

### 3. Môi trường Thử nghiệm Chấp nhận

Thiết bị:

* Tối thiểu 2 máy tính (Windows 10/11 hoặc macOS), 1 tablet (Android/iOS), 1 smartphone (Android/iOS).

Phần mềm:

* Trình duyệt Chrome, Firefox, Safari, Edge – bản mới nhất.
* Android từ phiên bản 12 trở lên, iOS từ phiên bản 16 trở lên.

Mạng:

* Kết nối Internet ổn định (có thể mô phỏng Wi-Fi và 4G/5G).

Dữ liệu kiểm thử:

* Tối thiểu 5 tài khoản hợp lệ.
* Các trường hợp không hợp lệ: định dạng sai, mật khẩu yếu, email trùng lặp.

### 4. Quy trình Tổng quan Thực hiện UAT

Chuẩn bị:

* Product Owner xây dựng kịch bản kiểm thử (Test Cases) theo từng User Story.
* Tạo dữ liệu mẫu và môi trường kiểm thử.
* Hướng dẫn sử dụng cơ bản cho người dùng đại diện.

Thực thi:

* Người dùng thực hiện kiểm thử trên các thiết bị và trình duyệt đã xác định.
* Ghi lại kết quả kiểm thử (Pass/Fail), thời gian phản hồi, các quan sát.

Ghi nhận & Báo cáo:

* Tất cả lỗi và đề xuất cải tiến được ghi nhận trên hệ thống quản lý lỗi (ví dụ: Jira).
* Bao gồm mô tả lỗi, bước tái hiện, mức độ nghiêm trọng, hình ảnh/video minh họa.
* Báo cáo UAT được cập nhật định kỳ cho toàn đội.

Đánh giá & Quyết định:

* Product Owner và đội phát triển họp đánh giá kết quả UAT.
* Dựa trên tiêu chí chấp nhận để ra quyết định: chấp nhận/tạm hoãn/từ chối.

### 5. Vai trò và Trách nhiệm

Người dùng cuối/Đại diện người dùng:

* Thực hiện kiểm thử theo hướng dẫn.
* Ghi nhận lỗi, góp ý cải tiến, đánh giá mức độ hài lòng.

Product Owner:

* Chuẩn bị kế hoạch UAT, phân công tester, tổng hợp kết quả và đưa ra khuyến nghị cuối cùng.
* Đội Phát triển:
* Hỗ trợ kỹ thuật, ghi nhận và xử lý các lỗi được báo cáo.

### 6. Tiêu chí Chấp nhận (Pass/Fail Criteria)

Hoàn thành Kịch bản Kiểm thử:

* 100% các kịch bản kiểm thử liên quan đến Đăng ký, Đăng nhập, và quản lý phiên làm việc phải Pass.
* Ít nhất 95% tổng số các kịch bản UAT trong Sprint 1 (bao gồm kiểm tra bảo mật, responsive UI trên nhiều thiết bị) phải Pass để được xem là chấp nhận được.

Quản lý Lỗi (Defect Management):

* 0 lỗi Mức độ Nghiêm trọng (Critical): Không được có lỗi gây mất dữ liệu người dùng, đăng nhập không thể thực hiện hoặc lộ thông tin cá nhân/mật khẩu.
* Tối đa 1 lỗi Mức độ Cao (High): Các lỗi ảnh hưởng đến trải nghiệm quan trọng (ví dụ: giao diện không responsive trên thiết bị phổ biến) và cần có kế hoạch sửa chữa cụ thể trước khi triển khai tiếp các Sprint sau.
* Tối đa 3 lỗi Mức độ Trung bình (Medium): Ví dụ như hiển thị sai font chữ hoặc lỗi layout nhẹ trên thiết bị ít phổ biến.
* Lỗi Mức độ Thấp (Low) như lỗi chính tả, căn lề lệch nhỏ có thể ghi nhận để xử lý trong Sprint sau nếu không ảnh hưởng đến luồng sử dụng.

Yêu cầu Phi Chức năng:

* Tính dễ sử dụng: Ít nhất 80% người dùng UAT đánh giá giao diện là “Dễ sử dụng” hoặc “Rất dễ sử dụng” thông qua khảo sát nhanh sau khi kiểm thử.
* Hiệu năng: Thời gian phản hồi của các thao tác xác thực (đăng nhập, đăng ký) phải < 1 giây. Trang giao diện hiển thị hoàn chỉnh trên thiết bị có tốc độ mạng trung bình < 2 giây.
* Responsive UI: Giao diện không được vỡ bố cục trên các màn hình thông dụng (mobile 5.5"-6.7", tablet, desktop >= 13").

Tài liệu:

* Tài liệu hướng dẫn cơ bản (hoặc màn hình trợ giúp) phải có sẵn và dễ hiểu, bao gồm ít nhất: đăng ký, đăng nhập, xử lý lỗi phổ biến.

### 7. Quy trình Ra Quyết định Chấp nhận

Product Owner, dựa vào báo cáo UAT, tỷ lệ Pass của các kịch bản và đánh giá mức độ nghiêm trọng của lỗi, sẽ đưa ra một trong các quyết định:

* Chấp nhận hoàn toàn: Nếu tất cả các tiêu chí trong 6 được đáp ứng.
* Chấp nhận có điều kiện: Nếu tồn tại lỗi mức cao hoặc trung bình nhưng đã có kế hoạch khắc phục cụ thể và không ảnh hưởng lớn đến khả năng phát triển tiếp.
* Từ chối: Nếu có lỗi nghiêm trọng hoặc phần lớn người dùng đánh giá trải nghiệm không đạt yêu cầu cơ bản.

Quyết định cuối cùng do Product Owner phê duyệt sau khi tham khảo ý kiến từ đại diện đội phát triển, QA và người dùng thử nghiệm (nếu cần).

### 8. Hình thức Xác nhận Chấp nhận

Một Biên bản Nghiệm thu Chấp nhận Người dùng (UAT Sign-off Document) cho Sprint 1 sẽ được lập, bao gồm:

* Tóm tắt các nội dung kiểm thử, tỷ lệ Pass/Fail, tổng quan lỗi phát hiện.
* Đánh giá tổng thể từ Product Owner và đại diện người dùng (nếu có).
* Chữ ký xác nhận của Product Owner, và tùy chọn của QA Lead và/hoặc người dùng đại diện.

Biên bản này là cơ sở để chuyển sang Sprint tiếp theo hoặc phát hành phiên bản nền tảng (foundation release).

## II. Sprint 2: Chức năng quản lý nội dung cơ bản

### 1. Mục đích

Sprint 2 của kiểm thử chấp nhận người dùng (UAT) nhằm xác minh các chức năng CRUD (Tạo, Đọc, Cập nhật, Xóa) cho bài viết, hình ảnh, và phân loại nội dung hoạt động chính xác, ổn định, và đáp ứng nhu cầu thực tế của người dùng. Mục tiêu chính là đảm bảo người dùng có thể tự chủ tạo, chỉnh sửa và tổ chức nội dung một cách dễ dàng và hiệu quả.

### 2. Phạm vi thử nghiệm chấp nhận

Phạm vi UAT bao gồm các chức năng sau:

* US02 – Tạo bài viết mới: Kiểm tra khả năng tạo bài viết với tiêu đề, nội dung rich text, chọn danh mục, đính kèm hình ảnh hoặc tệp. Đảm bảo bài viết được lưu và hiển thị đúng.
* US04 – Chỉnh sửa bài viết: Đánh giá khả năng chỉnh sửa nội dung, tiêu đề và hình ảnh/tệp đã đính kèm. Xác minh thay đổi được lưu chính xác và hiển thị đúng.
* US05 – Xóa bài viết: Kiểm tra chức năng xóa một hoặc nhiều bài viết, bao gồm hộp thoại xác nhận và việc bài viết biến mất khỏi hệ thống.
* US09 – Quản lý hình ảnh và tệp đính kèm: Xác minh khả năng tải lên, xem trước, chèn vào bài viết và xóa các định dạng tệp phổ biến như JPG, PNG, PDF, DOCX với nhiều kích thước.
* US06 – Phân loại bài viết theo danh mục: Kiểm tra khả năng tạo, chỉnh sửa, xóa danh mục và gán bài viết vào danh mục. Đảm bảo việc phân loại và hiển thị bài viết theo danh mục chính xác.
* US15.2 – Responsive cho chức năng quản lý nội dung: Kiểm tra giao diện quản lý nội dung trên desktop, tablet và smartphone, đặc biệt đánh giá trải nghiệm của trình soạn thảo rich text trên thiết bị di động.

### 3. Môi trường thử nghiệm chấp nhận

Phần cứng:

* Tối thiểu: 2 máy tính (PC/Laptop), 1 tablet (Android/iOS), 1 smartphone (Android/iOS).

Phần mềm:

* Trình duyệt: Chrome, Firefox, Edge, Safari (phiên bản mới nhất).
* Hệ điều hành: Windows, macOS, Android, iOS.

Mạng:

* Kết nối Internet ổn định.

Dữ liệu kiểm thử:

* Tối thiểu 10 bài viết mẫu với nội dung, định dạng phong phú.
* Tệp mẫu bao gồm các định dạng phổ biến (JPG, PNG, PDF, DOCX).
* Ít nhất 5 danh mục nội dung.
* Tài khoản người dùng có quyền quản lý nội dung.

### 4. Quy trình tổng quan thực hiện UAT

Chuẩn bị:

* Xây dựng kịch bản kiểm thử chi tiết cho từng chức năng.
* Chuẩn bị dữ liệu kiểm thử phong phú.
* Đảm bảo môi trường thử nghiệm ổn định và sẵn sàng.

Thực thi:

* Người dùng cuối thực hiện các kịch bản kiểm thử.
* Tập trung kiểm tra các chức năng tạo, chỉnh sửa, xóa bài viết và xử lý media.
* Kiểm thử trên nhiều thiết bị để đánh giá tính responsive.

Ghi nhận và báo cáo kết quả:

* Ghi nhận lỗi/khuyết điểm vào hệ thống quản lý lỗi (có mô tả, ảnh chụp/video, mức độ ưu tiên).
* Cập nhật tiến độ và báo cáo kết quả kiểm thử định kỳ.

Đánh giá và quyết định:

* Đội dự án cùng Product Owner xem xét kết quả kiểm thử.
* Đánh giá mức độ đáp ứng yêu cầu và ổn định chức năng.

### 5. Vai trò và trách nhiệm

Người dùng cuối/Đại diện người dùng:

* Thực hiện kiểm thử, đánh giá trải nghiệm và mức độ dễ sử dụng.

Product Owner:

* Xem xét kết quả, đánh giá chất lượng chức năng, đưa ra quyết định chấp nhận.

Quản lý dự án:

* Điều phối toàn bộ hoạt động UAT, đảm bảo đúng tiến độ.

Đội phát triển:

* Hỗ trợ kỹ thuật và xử lý lỗi được ghi nhận.

Kiểm thử viên (QA):

* Hướng dẫn người dùng kiểm thử, ghi nhận và xác minh lỗi.

### 6. Tiêu chí chấp nhận (Pass/Fail Criteria)

Hoàn thành kịch bản kiểm thử:

* Tỷ lệ hoàn thành kịch bản kiểm thử tổng thể: ≥ 95%.
* Tỷ lệ thành công bắt buộc cho các chức năng chính:
* US02, US04, US05: 100%.
* US09, US06: ≥ 98%.

Quản lý lỗi:

* Lỗi Critical (Nghiêm trọng): Không được tồn tại.
* Lỗi High (Cao): Tối đa 2 lỗi, có kế hoạch khắc phục rõ ràng.
* Lỗi Medium (Trung bình): Tối đa 5 lỗi, có kế hoạch xử lý cụ thể.
* Lỗi Low (Thấp): Có thể tồn tại, nếu đã được chấp thuận chính thức hoặc có kế hoạch khắc phục.

Yêu cầu phi chức năng:

* Thời gian tải giao diện tạo/chỉnh sửa bài viết: ≤ 4 giây.
* Thời gian lưu bài viết hoặc upload tệp: ≤ 6 giây.
* Trình soạn thảo Rich Text hoạt động ổn định, không bị treo, không mất dữ liệu.
* Tất cả giao diện quản lý nội dung hiển thị đúng trên ≥ 95% thiết bị và trình duyệt được kiểm thử.

Tài liệu hỗ trợ:

* Có tài liệu hướng dẫn chi tiết và dễ hiểu cho:
* Tạo, chỉnh sửa, xóa bài viết.
* Quản lý hình ảnh/tệp đính kèm.
* Quản lý danh mục nội dung.

### 7. Quy trình ra quyết định chấp nhận

Sau khi hoàn tất kiểm thử, Product Owner sẽ xem xét toàn bộ kết quả UAT (báo cáo, lỗi còn tồn tại, tài liệu hướng dẫn, mức độ hoàn thiện). Dựa trên đánh giá này, PO sẽ đưa ra một trong ba quyết định:

* Chấp nhận hoàn toàn: Đầy đủ tiêu chí và không có lỗi nghiêm trọng tồn tại.
* Chấp nhận có điều kiện: Một số lỗi nhỏ/tài liệu còn thiếu nhưng có thể bổ sung sau khi triển khai.
* Từ chối: Không đáp ứng tiêu chí, cần phát triển/kiểm thử lại.

Product Owner là người có quyền quyết định cuối cùng.

### 8. Hình thức xác nhận chấp nhận

Việc xác nhận chấp nhận UAT được thực hiện thông qua Biên bản nghiệm thu người dùng (User Acceptance Test Sign-off Document), có đầy đủ:

* Chữ ký của Product Owner.
* Chữ ký của Quản lý dự án.
* Thời gian và phạm vi chấp nhận cụ thể.

## III. Sprint 3: Trải nghiệm người dùng nâng cao và AI

### 1. Mục đích

UAT Sprint 3 nhằm xác nhận các chức năng xem trước bài viết, tìm kiếm, lọc, sắp xếp và đặc biệt là tính năng tạo nội dung bằng AI hoạt động hiệu quả, chính xác, đồng thời mang lại trải nghiệm người dùng tốt nhất trên mọi thiết bị. Mục tiêu là đảm bảo người dùng có công cụ mạnh mẽ để quản lý và tạo ra nội dung một cách thông minh và tiện lợi.

### 2. Phạm vi Thử nghiệm chấp nhận

Phạm vi UAT Sprint 3 tập trung vào các tính năng chính:

* US16 - Chế độ xem trước bài viết: Kiểm tra khả năng xem trước bài viết trong quá trình soạn thảo hoặc trước khi đăng, đảm bảo hiển thị đúng định dạng, nội dung, hình ảnh trên các thiết bị.
* US10 - Tìm kiếm bài viết: Đánh giá khả năng tìm kiếm theo từ khóa (tiêu đề, nội dung), hệ thống trả về kết quả nhanh chóng và chính xác. Kiểm thử cả trường hợp không có kết quả hoặc từ khóa đặc biệt.
* US07 - Lọc và sắp xếp bài viết: Xác minh khả năng lọc bài viết theo danh mục, ngày đăng, trạng thái và sắp xếp theo nhiều tiêu chí (ngày mới nhất/cũ nhất, A-Z/Z-A, lượt xem), đảm bảo hoạt động chính xác khi sử dụng độc lập hoặc kết hợp.
* US03 - Tự động tạo nội dung AI: Kiểm tra khả năng người dùng cung cấp từ khóa/chủ đề và hệ thống tạo nội dung văn bản. Đánh giá chất lượng nội dung AI về tính liên quan, chính xác, mạch lạc, ngữ pháp và độ dài phù hợp. Kiểm thử các trường hợp nội dung không đạt yêu cầu.
* US15.3 - Giao diện tìm kiếm và lọc responsive: Đảm bảo thanh tìm kiếm và các bộ lọc hiển thị rõ ràng, dễ thao tác trên mọi thiết bị. Kiểm tra kết quả tìm kiếm và danh sách bài viết sau lọc/sắp xếp hiển thị tối ưu trên các kích thước màn hình khác nhau.

### 3. Môi trường thử nghiệm chấp nhận

Phần cứng:

* Tối thiểu 2 PC/Laptop, 1 Tablet (Android/iOS), 1 Smartphone (Android/iOS).

Phần mềm:

* Trình duyệt Chrome, Firefox, Edge, Safari phiên bản mới nhất; Hệ điều hành Windows/macOS, Android/iOS.

Mạng:

* Kết nối Internet ổn định, đủ băng thông cho API AI.

Dữ liệu:

* Tối thiểu 50 bài viết đã đăng với nội dung đa dạng để thử nghiệm tìm kiếm/lọc. Bộ từ khóa/chủ đề mẫu (đơn giản, phức tạp, chuyên biệt) để kiểm thử AI.

### 4. Quy trình tổng quan thực hiện UAT

Chuẩn bị:

* Xây dựng kịch bản kiểm thử chi tiết, đặc biệt chú trọng đánh giá chất lượng nội dung AI và hiệu quả tìm kiếm/lọc. Chuẩn bị dữ liệu kiểm thử đa dạng cho các yêu cầu AI. Đảm bảo môi trường UAT có kết nối ổn định với API AI.

Thực thi:

* Người dùng cuối thực hiện các chức năng xem trước bài viết, tìm kiếm, lọc, sắp xếp. Tập trung kiểm thử chức năng AI, đánh giá kết quả và ghi nhận phản hồi chi tiết về chất lượng nội dung AI. Kiểm tra tính năng responsive của giao diện tìm kiếm và lọc trên thiết bị di động.

Ghi nhận và báo cáo kết quả:

* Ghi nhận lỗi, vấn đề và đề xuất cải tiến vào hệ thống quản lý lỗi. Với AI, ghi rõ từ khóa/chủ đề đầu vào cùng nội dung AI tạo ra. Tổng hợp báo cáo tiến độ và kết quả UAT định kỳ, bao gồm đánh giá định tính về chất lượng AI.

Đánh giá và ra quyết định:

* Đội ngũ dự án và Product Owner xem xét kết quả, đặc biệt chú trọng đến sự hài lòng của người dùng về chất lượng AI và trải nghiệm tìm kiếm/lọc.

### 5. Vai trò và Trách nhiệm

Người dùng cuối/Đại diện người dùng:

* Thực hiện kiểm thử, đánh giá chất lượng các tính năng xem trước, tìm kiếm, lọc, sắp xếp và nội dung do AI tạo ra.

Product Owner:

* Đánh giá chất lượng nội dung AI có đáp ứng yêu cầu nghiệp vụ không, phê duyệt tiêu chí chấp nhận liên quan đến AI và đưa ra quyết định cuối cùng.

Quản lý dự án:

* Điều phối tổng thể UAT.

Đội phát triển:

* Khắc phục lỗi, điều chỉnh/tinh chỉnh mô hình AI nếu chất lượng không đạt yêu cầu.

Kiểm thử viên: Hỗ trợ thực hiện kiểm thử, ghi nhận lỗi và kiểm tra lại sau khi khắc phục.

### 6. Tiêu chí Chấp nhận (Pass/Fail Criteria)

Hoàn thành kịch bản kiểm thử:

* Tỷ lệ kịch bản kiểm thử thành công tổng thể ≥ 90%.
* Tỷ lệ kịch bản kiểm thử thành công cho US16, US10, US07 đạt 100%.
* Tỷ lệ tạo nội dung AI (US03) ≥ 80% trường hợp được đánh giá “có ý nghĩa” và “phù hợp” với từ khóa/chủ đề.

Quản lý lỗi:

* Lỗi Critical: 0 tồn tại.
* Lỗi High: Tối đa 2 tồn tại, có kế hoạch khắc phục rõ ràng.
* Lỗi Medium: Tối đa 5 tồn tại, có kế hoạch khắc phục hoặc chấp nhận.
* Lỗi Low: Có thể tồn tại nhưng phải có kế hoạch hoặc được chấp nhận chính thức.

Yêu cầu phi chức năng:

* Thời gian phản hồi tìm kiếm và lọc ≤ 3 giây.
* Thời gian tạo nội dung AI ≤ 15 giây (tùy độ phức tạp và tải API AI).
* Tỷ lệ tìm kiếm chính xác (Precision và Recall) ≥ 90%.
* Nội dung AI tạo ra mạch lạc, ngữ pháp đúng, độ dài phù hợp.
* Giao diện tìm kiếm và lọc hoạt động tốt trên ≥ 95% thiết bị đã kiểm thử.

Hoàn thiện tài liệu:

* Tài liệu hướng dẫn chi tiết cho các chức năng xem trước, tìm kiếm, lọc/sắp xếp và sử dụng AI tạo nội dung (cách cung cấp từ khóa hiệu quả, chỉnh sửa nội dung AI).

### 7. Quy trình ra quyết định chấp nhận

Product Owner sẽ đưa ra quyết định cuối cùng dựa trên kết quả UAT, đặc biệt cân nhắc chất lượng nội dung AI và khả năng đáp ứng nhu cầu người dùng. Việc đánh giá chất lượng AI có thể mang tính chủ quan và cần sự đồng thuận.

### 8. Hình thức xác nhận chấp nhận

Biên bản nghiệm thu chấp nhận người dùng (User Acceptance Test Sign-off Document) có chữ ký xác nhận của Product Owner và Quản lý dự án.

## IV. Sprint 4: SEO và Lên Lịch Tự Động

### 1. Mục đích

UAT Sprint 4 nhằm xác nhận các chức năng tối ưu hóa nội dung cho SEO, lên lịch đăng bài tự động, thông báo khi đăng bài thành công, và nhập/xuất dữ liệu hoạt động ổn định, chính xác và đáng tin cậy. Mục tiêu là đảm bảo người dùng có thể tự động hóa quy trình xuất bản, tối ưu hóa khả năng hiển thị trên công cụ tìm kiếm và quản lý dữ liệu linh hoạt.

### 2. Phạm vi thử nghiệm chấp nhận

Phạm vi UAT Sprint 4 bao gồm:

* US11 - Tối ưu hóa SEO tự động: Kiểm tra hệ thống phân tích nội dung và đưa ra đề xuất từ khóa, tiêu đề, meta description, mật độ từ khóa, liên kết nội bộ/bên ngoài. Đảm bảo người dùng dễ dàng xem và áp dụng đề xuất, đánh giá tính chính xác và hữu ích.
* US08 - Lên lịch đăng bài tự động: Xác minh người dùng có thể đặt lịch đăng bài vào thời điểm cụ thể, bài viết được đăng đúng lịch, đồng thời có thể xem lại, chỉnh sửa hoặc hủy lịch.
* US12 - Thông báo khi đăng bài thành công: Kiểm tra người dùng nhận được thông báo (email, trong ứng dụng) đầy đủ, kịp thời khi bài viết được đăng.
* US17 - Nhập/xuất dữ liệu:
  + *Xuất dữ liệu:* Kiểm tra xuất bài viết ra các định dạng PDF, Word (.docx), CSV, đảm bảo tính toàn vẹn về nội dung và định dạng.
  + *Nhập dữ liệu:* Kiểm tra nhập bài viết từ các định dạng hỗ trợ (ví dụ CSV), đảm bảo dữ liệu chính xác.
* US15.4 - Giao diện lên lịch và SEO responsive: Đảm bảo giao diện hiển thị rõ ràng, dễ thao tác trên desktop, tablet và mobile, các thành phần tương tác hoạt động tốt trên màn hình cảm ứng.

### 3. Môi trường thử nghiệm chấp nhận

Phần cứng:

* Tối thiểu 2 PC/Laptop, 1 Tablet (Android/iOS), 1 Smartphone (Android/iOS).

Phần mềm:

* Trình duyệt Chrome, Firefox, Edge, Safari phiên bản mới nhất; hệ điều hành Windows/macOS, Android/iOS; các phần mềm đọc/tạo tệp (Microsoft Word, Excel, Adobe Acrobat Reader).

Mạng:

* Kết nối Internet ổn định, cần thiết cho thông báo email.

Dữ liệu:

* Dữ liệu bài viết mẫu (đã tối ưu và chưa tối ưu SEO), bài viết mẫu đặt lịch, dữ liệu mẫu các định dạng (CSV, Word, PDF) dùng cho nhập/xuất.

### 4. Quy trình tổng quan thực hiện UAT

Chuẩn bị:

* Xây dựng kịch bản kiểm thử chi tiết cho các chức năng SEO, lên lịch đăng bài, nhập/xuất dữ liệu. Đảm bảo môi trường có khả năng gửi/nhận email.

Thực thi:

* Người dùng cuối thực hiện các tác vụ theo kịch bản; kiểm tra bài viết đăng đúng lịch, nhận thông báo, tính toàn vẹn dữ liệu sau nhập/xuất, kiểm tra tính responsive.

Ghi nhận và báo cáo:

* Ghi nhận lỗi, vấn đề, đề xuất cải tiến vào hệ thống quản lý lỗi; tổng hợp báo cáo tiến độ, kết quả UAT định kỳ.

Đánh giá và quyết định:

* Đội dự án và Product Owner xem xét kết quả, đặc biệt chú trọng độ tin cậy và ổn định các chức năng.

### 5. Vai trò và trách nhiệm

Người dùng cuối/Đại diện người dùng:

* Thực hiện kiểm thử, đánh giá tính chính xác, hữu ích và ổn định.

Product Owner:

* Đánh giá tổng thể, đảm bảo đáp ứng yêu cầu và đưa ra quyết định chấp nhận cuối cùng.

Quản lý dự án:

* Điều phối tổng thể UAT.

Đội phát triển:

* Khắc phục lỗi, hỗ trợ kỹ thuật.

Kiểm thử viên:

* Hỗ trợ thực hiện kiểm thử, ghi nhận và kiểm tra lại lỗi.

### 6. Tiêu chí chấp nhận (Pass/Fail Criteria)

Kịch bản kiểm thử:

* Tỷ lệ thành công tổng thể ≥ 95%.
* US08 (lên lịch đăng bài) đạt 100%.
* US12 (thông báo) và US17 (nhập/xuất dữ liệu) ≥ 98%.
* US11 (SEO): ≥ 90% đề xuất hợp lý và áp dụng được.

Quản lý lỗi:

* Lỗi Critical: 0 tồn tại.
* Lỗi High: Tối đa 1 lỗi tồn tại, phải có kế hoạch khắc phục rõ ràng.
* Lỗi Medium: Tối đa 3 lỗi tồn tại, có kế hoạch hoặc chấp nhận.
* Lỗi Low: Có thể tồn tại, phải có kế hoạch hoặc chấp nhận chính thức.

Yêu cầu phi chức năng:

* Độ tin cậy lên lịch: 100%.
* Thời gian xử lý nhập/xuất dữ liệu theo khối lượng.
* Thông báo đăng bài phải gửi trong vòng 5 phút.
* Giao diện responsive hoạt động tốt trên ≥ 95% thiết bị kiểm thử.

Tài liệu:

* Hướng dẫn sử dụng chi tiết cho các chức năng SEO, lên lịch, cấu hình thông báo, nhập/xuất dữ liệu.

### 7. Quy trình ra quyết định chấp nhận

Product Owner đưa ra quyết định cuối cùng dựa trên kết quả UAT, tập trung vào tính ổn định và độ tin cậy của các chức năng tự động hóa và quản lý dữ liệu.

### 8. Hình thức xác nhận chấp nhận

Biên bản nghiệm thu UAT có chữ ký xác nhận của Product Owner và Quản lý dự án.