

User Stories Sprint 1 (nên làm đầu tiên): Đăng nhập Đăng ký Tạo bài viết nhật ký Xem danh sách bài viết  
Xem chi tiết bài viết Chính sửa bài viết Xóa bài viết

Tên sinh viên	Phạm Nhật Tân
MSSV	D24DTCN443
Lớp	B24TXCN08-B
Tên dự án	Quản lý viết blog nhật ký
Ngày nộp	04/12/2025

## 1. Sprint Planning & Product Backlog

### 1.1 Sprint Goal

Hoàn thiện các chức năng cốt lõi cho hệ thống viết blog nhật ký, bao gồm xác thực người dùng và các thao tác cơ bản với bài viết (tạo, xem, sửa, xóa), nhằm giúp người dùng có thể đăng nhập và sử dụng đầy đủ quy trình quản lý bài viết.

### 1.2 Danh sách User Stories (5–7 stories)

STT	User Story	Mô tả ngắn	Story Point	Mức độ ưu tiên
1	EM-01	Cho phép người dùng tạo tài khoản mới để sử dụng hệ thống	3	Must
2	EM-02	Người dùng đăng nhập vào hệ thống bằng email/mật khẩu	3	Must
3	EM-03	Người dùng tạo một bài nhật ký mới và lưu vào hệ thống	5	Must
4	EM-04	Hiển thị toàn bộ bài viết theo thứ tự thời gian	3	Must
5	EM-05	Người dùng mở và xem nội dung	2	Must

		đầy đủ của một bài nhật ký		
6	EM-06	Cho phép chỉnh sửa nội dung bài viết đã tạo	5	Should
7	EM-07	Xóa một bài viết khỏi danh sách nhật ký	2	Should

## 1.2 Acceptance Criteria (AC) cho từng User Story

Tên Story	Mô tả tiêu chí (Acceptance Criteria)
<b>Đăng ký</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể nhập email hợp lệ và mật khẩu.</li> <li>- Hệ thống kiểm tra email chưa được đăng ký.- Hiển thị thông báo thành công khi đăng ký.</li> <li>- Hiển thị lỗi nếu dữ liệu không hợp lệ hoặc email đã tồn tại.</li> <li>- Mật khẩu được lưu an toàn (hash).</li> </ul>
<b>Đăng nhập</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể đăng nhập bằng email và mật khẩu.</li> <li>- Hiển thị thông báo lỗi nếu email hoặc mật khẩu sai.</li> <li>- Sau khi đăng nhập thành công, chuyển hướng đến trang dashboard.</li> <li>- Hệ thống giữ phiên đăng nhập (session/token).</li> </ul>
<b>Tạo bài viết nhật ký</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể nhập tiêu đề và nội dung bài viết.</li> <li>- Nội dung bài viết có thể định dạng cơ bản (bold, italic).</li> <li>- Bài viết được lưu vào hệ thống (database/localStorage).</li> <li>- Hiển thị thông báo thành công khi tạo bài.- Kiểm tra không cho tạo bài trùng.</li> </ul>
<b>Xem danh sách bài viết</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể thấy danh sách các bài viết đã tạo.</li> <li>- Mỗi bài viết hiển thị tiêu đề, ngày tạo, tóm tắt nội dung.</li> <li>- Có thể sắp xếp theo ngày tạo mới nhất hoặc cũ nhất.</li> <li>- Hỗ trợ phân trang nếu nhiều bài viết.</li> </ul>
<b>Xem chi tiết bài viết</b>	<ul style="list-style-type: none"> <li>- Người dùng nhấn vào bài viết để xem toàn bộ nội dung.</li> <li>- Hiển thị tiêu đề, nội dung, ngày tạo, tác giả.</li> </ul>

	<ul style="list-style-type: none"> <li>- Cho phép quay lại danh sách bài viết.</li> <li>- Nếu bài viết không tồn tại, hiển thị thông báo lỗi.</li> </ul>
<b>Chỉnh sửa bài viết</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể chỉnh sửa tiêu đề và nội dung bài viết.</li> <li>- Kiểm tra không cho lưu bài trống.</li> <li>- Lưu lại thay đổi và cập nhật danh sách bài viết.</li> <li>- Hiển thị thông báo thành công hoặc lỗi.</li> </ul>
<b>Xóa bài viết</b>	<ul style="list-style-type: none"> <li>- Người dùng có thể xóa bài viết từ danh sách.</li> <li>- Hiển thị xác nhận trước khi xóa.</li> <li>- Sau khi xóa, bài viết biến mất khỏi danh sách.</li> <li>- Hiển thị thông báo thành công hoặc lỗi nếu xóa không thành công.</li> </ul>

### 1.3 Definition of Done (DoD)

STT	Định nghĩa hoàn thành (Definition of Done)
1	Backlog hoàn chỉnh, tất cả User Story được viết đầy đủ tiêu chí chấp nhận (Acceptance Criteria).
2	Mã nguồn đã được viết, chạy ổn định, không còn lỗi nghiêm trọng.
3	Code đã được review và merge vào nhánh chính (main/master).
4	Tất cả các tính năng đã được kiểm thử (unit test / functional test) và pass.
5	Tài liệu liên quan (hướng dẫn sử dụng, comment code) đã được cập nhật đầy đủ.
6	Chức năng được triển khai đúng theo yêu cầu người dùng và đáp ứng Acceptance Criteria.
7	Sprint Review đã diễn ra và Product Owner xác nhận hoàn thành.

### 1.4 Estimation – Planning Poker

Story	Story Point	Lý do chọn điểm
Đăng ký	3	Form đăng ký đơn giản, chỉ cần kiểm tra email hợp lệ, password hợp lệ, lưu dữ liệu.
Đăng nhập	3	Tương tự đăng ký, cần xác thực email/password và quản lý session.

Tạo bài viết nhật ký	3	Cần thiết kế form, validate nội dung, lưu dữ liệu, hiển thị thông báo thành công.
Xem danh sách bài viết	3	Phải render danh sách, hỗ trợ phân trang, sắp xếp theo ngày tạo, hiển thị tóm tắt.
Xem chi tiết bài viết	3	Chỉ cần hiển thị nội dung chi tiết, tiêu đề, ngày tạo, xử lý trường hợp bài không tồn tại.
Chỉnh sửa bài viết	5	Form chỉnh sửa, validate dữ liệu, lưu thay đổi, cập nhật danh sách bài viết.
Xóa bài viết	3	Thêm xác nhận xóa, xóa dữ liệu khỏi danh sách, hiển thị thông báo thành công hoặc lỗi.

## 2. Kanban Board

### 2.1 Chia nhỏ User Story thành Task

Story	Task	Người thực hiện	Mô tả
EM-01	EM-01-T1	Dev	Tạo giao diện form đăng ký (UI: email, mật khẩu, xác nhận mật khẩu).
	EM-01-T2	Dev	Xử lý logic đăng ký: validate dữ liệu, lưu tài khoản vào database/localStorage.
	EM-01-T3	Tester	Kiểm thử các trường hợp đăng ký: thành công, trùng email, nhập sai định dạng.
EM-02	EM-02-T1	Dev	Tạo giao diện trang đăng nhập.
	EM-02-T2	Dev	Viết logic đăng nhập: kiểm tra email/mật khẩu, lưu session.
	EM-02-T3	Tester	Test đăng nhập đúng/sai, test lỗi giao diện, test xử lý session.
EM-03	EM-03-T1	Dev	Tạo giao diện form tạo bài viết: tiêu đề, nội dung, ngày tạo.
	EM-03-T2	Dev	Viết logic lưu bài viết vào hệ thống/localStorage.

	EM-03-T3	Tester	Kiểm thử tạo bài viết: trường rỗng, lưu thành công, giao diện hiển thị.
EM-04	EM-04-T1	Dev	Render danh sách bài viết từ hệ thống lên UI.
	EM-04-T2	Dev	Sắp xếp danh sách theo thời gian mới nhất.
	EM-04-T3	Tester	Kiểm thử hiển thị danh sách, thứ tự, số lượng bài viết.
EM-05	EM-05-T1	Dev	Tạo giao diện trang xem chi tiết bài viết.
	EM-05-T2	Dev	Viết logic lấy đúng bài viết theo ID và hiển thị đầy đủ.
	EM-05-T3	Tester	Test điều hướng, test nội dung tải đúng, test lỗi load dữ liệu.
EM-06	EM-06-T1	Dev	Tạo giao diện chỉnh sửa bài viết, pre-fill dữ liệu.
	EM-06-T2	Dev	Viết logic cập nhật bài viết vào hệ thống.
	EM-06-T3	Tester	Kiểm thử chỉnh sửa: cập nhật thành công, refresh lại dữ liệu, test lỗi nhập.
EM-07	EM-07-T1	Dev	Thêm nút xóa bài viết và hiển thị popup xác nhận.
	EM-07-T2	Dev	Viết logic xóa khỏi hệ thống/localStorage.
	EM-07-T3	Tester	Test xóa thành công, test hủy xóa, test danh sách cập nhật đúng.

## 2.2 Kanban Board

Ngày 1

User Story	Todo	In Progress	Pending	Review	Done
EM-01	EM01-T2, EM01-T3	EM01-T1: Thiết kế form đăng ký Start: d1 Due: d2			

<b>EM-02 -&gt; EM-07</b>	Tất cả task				
--------------------------	-------------	--	--	--	--

- Thay đổi:
  - Bắt đầu EM01-T1: Thiết kế UI form đăng ký.
- Lý do:
  - Tập trung vào UI vì đây là bước nền tảng trước khi làm logic.
  - EM01-T1 được làm trước để chuẩn bị cho toàn bộ flow xác thực.
  - Các user story khác chưa bắt đầu vì cần hoàn thiện form cơ bản trước.

Ngày 2:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-01</b>	EM01-T2, EM01-T3	EM01-T2: Xử lý logic  -Start: d2 -Due: d3	EM01-T1 Thiết kế form đăng ký  -Start: d2 -Due: d3		
<b>EM-02</b>	EM02-T2, EM02-T3	EM02-T1: Form login  -Start: d2 -Due: d3			
<b>EM-03 → EM07</b>	Tất cả task				

- Thay đổi:
  - Bắt đầu EM01-T2: Xử lý logic đăng ký.
  - Bắt đầu EM02-T1: Thiết kế UI form đăng nhập.
- Lý do:
  - Khi UI đăng ký xong → bắt đầu triển khai logic (EM01-T2).
  - Song song mở task UI đăng nhập (EM02-T1) vì không phụ thuộc vào logic đăng ký.

Ngày 3:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-01</b>		EM01-T3: Kiểm thử  -Start: d3 -Due: d4		EM01-T2 -Start: d3 -Due: d4	EM01-T1 -Start: d2 -Due: d3
<b>EM-02</b>	EM02-T3	EM02-T2: Xác			EM02-T1

		thực -Start: d3 -Due: d4			-Start: d2 -Due: d3
<b>EM-03</b>	EM03-T2, EM03-T3	EM03-T1: UI viết bài -Start: d3 -Due: d4			
<b>EM-04 → EM07</b>	Tất cả task				

- Thay đổi:
  - Bắt đầu EM01-T3: Kiểm thử đăng ký.
  - Bắt đầu EM02-T2: Xử lý logic xác thực đăng nhập.
  - Bắt đầu EM03-T1: Thiết kế UI viết bài.
- Lý do:
  - Đăng ký: chuyển sang kiểm thử (EM01-T3) vì logic đã hoàn thiện.
  - Đăng nhập: Xử lý xác thực (EM02-T2) do đã xong phần UI.
  - Tạo bài viết: bắt đầu UI (EM03-T1) vì đây là bước đầu tiên để viết bài.

#### Ngày 4:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-01</b>					EM01 Hoàn tất -Start: d1 -Due: d4
<b>EM-02</b>		EM02-T3: Kiểm thử -Start: d4 -Due: d4			EM02 Hoàn tất -Start: d2 -Due: d4
<b>EM-03</b>	EM03-T3	EM03-T2: Lưu dữ liệu -Start: d4 -Due: d5			EM03-T1 -Start: d3 -Due: d4
<b>EM-04</b>	EM04-T2, EM04-T3	EM04-T1 UI -Start: d4 -Due: d5			
<b>EM-05 → EM07</b>	Tất cả task				

- Thay đổi:
  - Bắt đầu EM02-T3: Kiểm thử đăng nhập.
  - Bắt đầu EM03-T2: Lưu dữ liệu bài viết.
  - Bắt đầu EM04-T1: Thiết kế UI chỉnh sửa/xóa bài viết.
- Lý do:
  - EM02 có thể kiểm thử độc lập.
  - Tao bài viết chuyển sang xử lý dữ liệu (EM03-T2) vì UI đã hoàn chỉnh.
  - Chỉnh sửa/Xóa bắt đầu UI (EM04-T1), logic xóa/update để ngày sau.

Ngày 5:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-01</b>					Hoàn tất
<b>EM-02</b>					Hoàn tất
<b>EM-03</b>		EM03-T3: Kiểm thử -Start: d5 -Due: d5			EM03 Hoàn tất -Start: d3 -Due: d5
<b>EM-04</b>	EM04-T3	EM04-T2: Update/Xóa -Start: d5 -Due: d7			EM04-T1 -Start: d4 -Due: d5
<b>EM-05</b>	EM05-T2, EM05-T3	EM05-T1: Gắn tag -Start: d5 -Due: d6			
<b>EM-06/07</b>	Tất cả task				

- Thay đổi:
  - Bắt đầu EM03-T3: Kiểm thử tạo bài viết.
  - Bắt đầu EM04-T2: Xử lý logic Update/Xóa bài viết.
  - Bắt đầu EM05-T1: Thiết kế UI gắn tag.
- Lý do:
  - EM03 logic có thể làm song song với các story khác.
  - Chỉnh sửa bài viết bắt đầu xử lý update/delete (EM04-T2).
  - Phân loại bắt đầu từ việc gắn tag (EM05-T1) — bước UI đầu tiên.

Ngày 6:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-01</b>					Hoàn tất
<b>EM-02</b>					Hoàn tất
<b>EM-03</b>					Hoàn tất

<b>EM-04</b>		EM04-T3: Kiểm thử CRUD -Start: d6 -Due: d7			Hoàn tất
<b>EM-05</b>	EM05-T3	EM05-T2: Lưu tag -Start: d6 -Due: d7			EM05-T1 -Start: d5 -Due: d6
<b>EM-06</b>	EM06-T2, EM06-T3	EM06-T1 -Start: d6 -Due: d7			
<b>EM-07</b>	Tất cả task				

- Thay đổi:
  - Bắt đầu EM04-T3: Kiểm thử CRUD (Chỉnh sửa/Xóa).
  - Bắt đầu EM05-T2: Xử lý logic lưu tag.
  - Bắt đầu EM06-T1: Xử lý logic truy vấn danh sách bài viết.
- Lý do:
  - EM04 bắt đầu kiểm thử vì UI và logic đã xong.
  - Phân loại chuyển sang xử lý lưu tag (EM05-T2) sau khi UI hoàn tất.
  - EM04 logic không phụ thuộc các story khác nên bắt đầu sớm.
  - Xem danh sách bắt đầu xử lý truy vấn (EM06-T1).

Ngày 7:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-05</b>		EM05-T3: Kiểm thử -Start: d7 -Due: d7			Hoàn tất
<b>EM-06</b>		EM06-T2: Hiển thị list -Start: d7 -Due: d8		EM06-T1 -Start: d7 -Due: d9	
<b>EM-07</b>	EM07-T2, EM07-T3	EM07-T1: Tích hợp nút chia sẻ -Start: d7 -Due: d8			

<b>EM01 → EM04</b>					Hoàn tất
------------------------	--	--	--	--	----------

- Thay đổi:
  - Bắt đầu EM05-T3: Kiểm thử phân loại tag.
  - Bắt đầu EM06-T2: Xử lý logic hiển thị danh sách bài viết.
  - Bắt đầu EM07-T1: Thiết kế UI tích hợp nút chia sẻ.
- Lý do:
  - Phân loại chuyển sang kiểm thử (EM05-T3) vì lưu dữ liệu đã hoàn thiện.
  - Xem danh sách bắt đầu xử lý hiển thị (EM06-T2) sau khi có logic truy vấn.
  - Chia sẻ bài viết bắt đầu tích hợp UI nút share (EM07-T1).

Ngày 8:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-06</b>		EM06-T3: Kiểm thử -Start: d8 -Due: d9		EM06-T2 -Start: d8 -Due: d9	
<b>EM-07</b>	EM07-T3	EM07-T2: Kết nối API -Start: d8 -Due: d9		EM07-T1 -Start: d8 -Due: d10	
<b>EM01 → EM05</b>					Hoàn tất

- Thay đổi:
  - Bắt đầu EM06-T3: Kiểm thử xem danh sách.
  - Bắt đầu EM07-T2: Xử lý logic kết nối API chia sẻ.
- Lý do:
  - Xem danh sách: kiểm thử (EM06-T3) sau khi hiển thị xong.
  - Chia sẻ bài viết làm kết nối API (EM07-T2) vì UI đã xong.

Ngày 9:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-07</b>	EM07-T3	EM07-T3: Kiểm thử chia sẻ -Start: d9 -Due: d10		EM07-T2 -Start: d9 -Due: d10	
<b>EM01 → EM06</b>					Hoàn tất

- Thay đổi:

- Bắt đầu EM07-T3: Kiểm thử chia sẻ bài viết.
- Lý do:
  - Tiến hành kiểm thử chia sẻ (EM07-T3) sau khi kết nối API xong.

Ngày 10:

User Story	Todo	In Progress	Pending	Review	Done
<b>EM-07</b>				EM07-T3: Kiểm thử chia sẻ -Start: d10 -Due: d10	<b>Hoàn tất</b>
<b>Tất cả user stories khác</b>					<b>Hoàn tất</b>

- Thay đổi:
  - Hoàn tất kiểm thử cho EM07 (Chia sẻ bài viết).
- Lý do:
  - Story cuối cùng (EM07) hoàn thành kiểm thử → Done.
  - Toàn bộ user story chuyển về Done, sprint kết thúc đúng phạm vi.
  - Sản phẩm đạt đủ yêu cầu: tạo, xem, sửa, xóa, phân loại, chia sẻ bài viết.

### 3. Phân tích rủi ro & Ưu tiên backlog

#### 3.1 Phân tích rủi ro

Story	Mức rủi ro	Lý do
<b>Story 1 – EM01: Xây UI trang Employee Management</b>	<b>Medium</b>	UI có nhiều thành phần (layout, table, modal), dễ xảy ra lỗi cần chỉnh, độ ưu tiên cao nên áp lực thời gian.
<b>Story 2 – EM02: Thêm mới nhân viên (Add Employee)</b>	<b>High</b>	Nhiều trường cần validate → dễ sai logic, liên quan dữ liệu và modal → dễ lỗi UX và sai format input.
<b>Story 3 – EM03: Chính sửa thông tin nhân viên (Edit Employee)</b>	<b>High</b>	Phụ thuộc dữ liệu tồn tại, xử lý cập nhật phức tạp hơn thêm mới, cần đảm bảo đồng bộ localStorage.
<b>Story 4 – EM04: Xoá nhân viên (Delete Employee)</b>	<b>Medium</b>	Rủi ro mất dữ liệu vĩnh viễn, cần confirm popup, nhưng logic đơn giản hơn Add/Edit.
<b>Story 5 – EM05: Tìm kiếm nhân viên (Search)</b>	<b>Low</b>	Chỉ lọc theo từ khoá text, không ảnh hưởng cấu trúc dữ liệu.

<b>Story 6 – EM06: Lọc nhân viên theo phòng ban (Filter)</b>	<b>Low</b>	Logic lọc đơn giản, ít rủi ro, không thay đổi dữ liệu.
<b>Story 7 – EM07: Sắp xếp nhân viên theo tiêu chí (Sort)</b>	<b>Medium</b>	Sắp xếp có thể gây thay đổi thứ tự hiển thị làm người dùng hiểu nhầm dữ liệu bị thay đổi vĩnh viễn. Cần xử lý rõ ràng hướng hiển thị.

### 3.2 MoSCoW

Story	Must	Should	Could	Won't	Lý do
<b>Story 1 – EM01: UI trang Employee Management</b>	<b>X</b>				UI là phần nền tảng, phải hoàn thiện trước để các chức năng khác hoạt động.
<b>Story 2 – EM02: Thêm mới nhân viên (Add Employee)</b>	<b>X</b>				Chức năng cốt lõi của hệ thống quản lý, ảnh hưởng trực tiếp đến dữ liệu.
<b>Story 3 – EM03: Chính sửa thông tin nhân viên (Edit Employee)</b>	<b>X</b>				Chỉnh sửa là chức năng chính, cần đầy đủ trước khi hệ thống vận hành thực tế.
<b>Story 4 – EM04: Xoá nhân viên (Delete Employee)</b>		<b>X</b>			Quan trọng nhưng không bắt buộc ngay sprint đầu; có thể hoàn thiện sau khi Add/Edit xong.
<b>Story 5 – EM05: Tìm kiếm nhân viên (Search)</b>		<b>X</b>			Giúp cải thiện trải nghiệm nhưng không ảnh hưởng hoạt động chính.
<b>Story 6 – EM06: Lọc nhân viên theo phòng ban (Filter)</b>			<b>X</b>		Là chức năng mở rộng, không bắt buộc; giúp tối ưu nhưng

					không cấp thiết.
<b>Story 7 – EM07: Sắp xếp nhân viên (Sort)</b>			X		Chỉ cải thiện UX, không phải tính năng cốt lõi trong Sprint đầu.

## 4. Mô phỏng phát triển và kiểm thử

### 4.1 Mô phỏng quá trình Dev (không code)

#### EM-01 – Đăng ký tài khoản

##### EM01-T1: Thiết kế form đăng ký

###### Mô phỏng:

- Dev tạo giao diện gồm: Họ tên, email, mật khẩu, xác nhận mật khẩu.
- Khi user nhập 1 trường → UI tự validate (email đúng định dạng).
- Nút “Đăng ký” chỉ sáng khi điền đúng hết.

##### EM01-T2: Xử lý logic đăng ký

###### Mô phỏng:

- User bấm “Đăng ký”.
- Hệ thống kiểm tra trùng email → nếu trùng báo lỗi.
- Nếu hợp lệ → tạo tài khoản và lưu vào DB giả lập.
- Trả về thông báo “Đăng ký thành công”.

##### EM01-T3: Kiểm thử đăng ký

###### Mô phỏng:

- Tester nhập email sai → hệ thống báo lỗi.
- Nhập email trùng → hệ thống không cho đăng ký.
- Nhập đầy đủ → tài khoản được thêm vào danh sách user.

### EM-02 – Đăng nhập

#### EM02-T1: Tạo form đăng nhập

###### Mô phỏng:

- Dev dựng UI gồm: Email, mật khẩu.
- Khi nhập thiếu → hệ thống báo “Không được để trống”.

##### EM02-T2: Xác thực thông tin

###### Mô phỏng:

- User nhập email/mật khẩu → hệ thống đối chiếu DB.

- Nếu sai → báo lỗi “Email hoặc mật khẩu không đúng”.
- Nếu đúng → chuyển sang Dashboard.

### **EM02-T3: Kiểm thử đăng nhập**

#### **Mô phỏng:**

- Tester thử đăng nhập sai nhiều lần → hệ thống vẫn chạy ổn.
- Nhập đúng → đăng nhập thành công.
- Kiểm tra redirect đúng trang.

### **EM-03 – Tạo bài viết**

#### **EM03-T1: Giao diện tạo bài viết**

#### **Mô phỏng:**

- Dev tạo UI gồm: tiêu đề, nội dung, ảnh đính kèm.
- Khi user nhập nội dung → auto hiển thị preview.

#### **EM03-T2: Lưu nội dung bài viết**

#### **Mô phỏng:**

- Khi user bấm “Đăng bài”:
  - Hệ thống kiểm tra trống.
  - Lưu bài viết vào DB → tạo ID mới.
  - Trả về “Tạo bài viết thành công”.

#### **EM03-T3: Kiểm thử tạo bài viết**

#### **Mô phỏng:**

- Tester thử tạo bài với tiêu đề rỗng → hệ thống từ chối.
- Tạo bài hợp lệ → xuất hiện trong danh sách bài viết.
- Kiểm tra ảnh upload có vào DB.

### **EM-04 – Chính sửa / Xóa bài viết**

#### **EM04-T1: UI chỉnh sửa / xóa**

#### **Mô phỏng:**

- Tạo trang chỉnh sửa với các trường cần thay đổi.
- Khi user nhấn icon “Delete” → hiện popup xác nhận.

#### **EM04-T2: Logic chỉnh sửa / xóa**

#### **Mô phỏng:**

- Sửa bài → hệ thống ghi đè dữ liệu trong DB.
- Xóa bài → hệ thống xóa theo ID.

- Sau thao tác → reload danh sách bài viết.

### **EM04-T3: Kiểm thử CRUD**

#### **Mô phỏng:**

- Tester sửa bài → kiểm tra DB thay đổi đúng.
- Xóa bài → bài biến mất khỏi danh sách.
- Undo → hệ thống không crash.

### **EM-05 – Phân loại bài viết (Tag & Category)**

#### **EM05-T1: Gắn tag/chủ đề**

#### **Mô phỏng:**

- Khi user nhập tag → gợi ý tag có sẵn.
- Chọn 1 tag → thêm vào list tag của bài viết.

#### **EM05-T2: Lưu tag vào DB**

#### **Mô phỏng:**

- Khi đăng bài → hệ thống gán tag vào bài viết.
- Tag được lưu dưới dạng array trong DB.
- Tự động tạo tag mới nếu chưa tồn tại.

#### **EM05-T3: Kiểm thử lọc theo tag**

#### **Mô phỏng:**

- Tester bấm tag “JavaScript” → danh sách chỉ còn bài JS.
- Nếu tag không có bài → hiển thị “Không có dữ liệu”.

### **EM-06 – Xem danh sách bài viết**

#### **EM06-T1: Truy vấn theo thời gian**

#### **Mô phỏng:**

- Khi user vào trang danh sách → hệ thống lấy bài viết theo thời gian mới nhất.
- Hiển thị 10 bài / trang (pagination giả lập).

#### **EM06-T2: Hiển thị danh sách**

#### **Mô phỏng:**

- Từng bài viết được render: ảnh, tiêu đề, mô tả ngắn.
- Nếu không có bài → hiện giao diện empty state.

#### **EM06-T3: Kiểm thử danh sách**

### **Mô phỏng:**

- Tester thêm mới bài → kiểm tra danh sách cập nhật.
- Kiểm tra phân trang hoạt động đúng.

### **EM-07 – Chia sẻ bài viết**

#### **EM07-T1: Tích hợp nút chia sẻ**

### **Mô phỏng:**

- Thêm button “Share to Facebook/Zalo”.
- Khi bấm → hiện popup xác nhận chia sẻ.

#### **EM07-T2: Kết nối API mạng xã hội**

### **Mô phỏng:**

- Khi user đồng ý → hệ thống gửi bài viết sang API giả lập.
- Nếu API lỗi → báo “Không thể chia sẻ, thử lại”.

#### **EM07-T3: Kiểm thử chia sẻ**

### **Mô phỏng:**

- Tester chia sẻ bài → kiểm tra trạng thái báo “Thành công”.
- Tắt mạng → hệ thống báo lỗi đúng như mong đợi.

## **4.2 Các kịch bản thử nghiệm**

Test Case	Input	Action	Expected Result
<b>TC01 – Đăng ký thành công</b>	Email hợp lệ, mật khẩu ≥ 6 ký tự	Nhập form → Register	Tài khoản được tạo, chuyển sang Login
<b>TC02 – Đăng ký thất bại (Email trùng)</b>	Email đã tồn tại	Register	Báo lỗi “Email already exists”
<b>TC03 – Đăng nhập thành công</b>	Email + mật khẩu đúng	Login	Chuyển vào Dashboard
<b>TC04 – Đăng nhập thất bại</b>	Mật khẩu sai	Login	Báo lỗi “Invalid password”
<b>TC05 – Tạo bài viết mới</b>	Tiêu đề + nội dung	Submit	Bài viết được lưu và hiển thị
<b>TC06 – Thiếu tiêu đề khi tạo bài viết</b>	Tiêu đề trống	Submit	Báo lỗi “Title is required”

<b>TC07 – Xem danh sách bài viết</b>	Không cần input	Mở trang danh sách	Hiển thị đầy đủ bài viết
<b>TC08 – Chính sửa bài viết</b>	Tiêu đề & nội dung mới	Edit → Save	Bài viết cập nhật thành công
<b>TC09 – Gắn tag bài viết</b>	Tag: "Học tập"	Chọn tag → Save	Tag hiển thị trong bài viết
<b>TC10 – Xóa bài viết</b>	ID bài viết	Delete	Bài viết biến mất khỏi danh sách
<b>TC11 – Chia sẻ bài viết</b>	URL share hợp lệ	Share	Tạo link chia sẻ thành công
<b>TC12 – Kiểm thử UI</b>	Không cần input	Load trang	UI hiển thị đúng layout
<b>TC13 – Test API chia sẻ</b>	GET /share?id=1	Gửi request	Trả về JSON bài viết

### 4.3 Bug giả lập

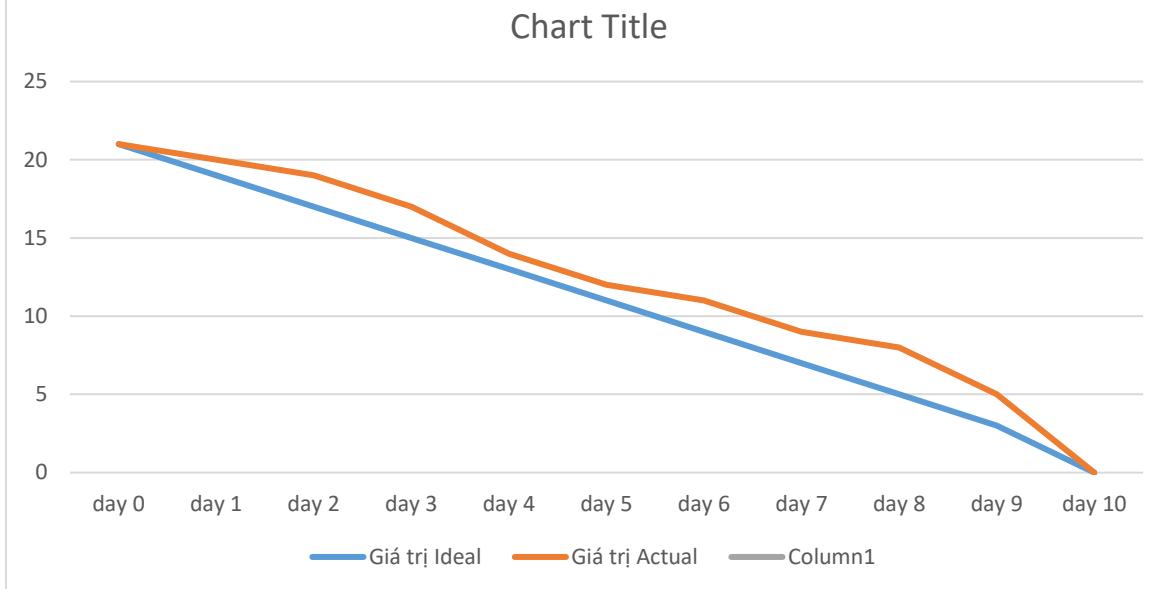
Bug ID	Mô tả lỗi	Mức độ nghiêm trọng	Các bước để tái hiện & giải quyết	Kết quả
B01	Không thể đăng ký khi email hợp lệ nhưng hệ thống báo "Email invalid"	Medium	1. Mở trang Register 2. Nhập email hợp lệ (vd: <a href="mailto:test@gmail.com">test@gmail.com</a> ) 3. Nhấn Register 4. Kiểm tra hàm validate → phát hiện regex sai → Dev sửa regex → Tester retest	Đăng ký hoạt động bình thường
B02	Đăng nhập không chuyển sang Dashboard mặc dù thông tin đúng	High	1. Nhập email + password đúng 2. Login → UI đúng yên 3. Kiểm tra logic redirect → thiếu navigate() 4. Dev bổ sung redirect	Hệ thống điều hướng đúng
B03	Không hiển thị bài viết sau khi tạo	High	1. Tạo bài viết mới 2. Submit thành	Bài viết xuất hiện trong danh sách

	mới		công nhưng list không update 3. Kiểm tra state → thiếu reload danh sách 4. Dev thêm hàm render lại	
B04	Chỉnh sửa bài viết nhưng nội dung cũ không được cập nhật	Medium	1. Mở bài viết → Edit 2. Save nhưng UI không đổi 3. Kiểm tra data binding → thiếu cập nhật localStorage 4. Dev fix logic update	Nội dung hiển thị mới
B05	Xóa bài viết nhưng refresh trang bài viết xuất hiện lại	High	1. Delete bài viết 2. Reload trang → bài viết quay lại 3. Kiểm tra → xóa chỉ trên UI, không xóa localStorage 4. Dev sửa để thực hiện xóa dữ liệu	Xóa hoàn toàn và không xuất hiện
B06	Tính năng gắn tag không lưu	Low	1. Chọn tag → Save 2. Reload → tag biến mất 3. Kiểm tra → thiếu push vào object bài viết 4. Dev cập nhật saveTag()	Tag được lưu và hiển thị
B07	Chia sẻ bài viết nhưng API trả lỗi 404	High	1. Nhấn Share 2. API /share?id=... trả 404 3. Kiểm tra → missing route 4. Dev thêm route mock API	Share hoạt động
B08	UI bị vỡ layout trên mobile	Medium	1. Mở giao diện trên màn hình nhỏ 2. Một số box bị tràn 3. Dev thêm responsive CSS	Layout hiển thị chuẩn

## 5. Tiết độ Sprint

### 5.1 Burndown Chart

Ngày	0	1	2	3	4	5	6	7	8	9	10
Ideal	21	19	17	15	13	11	9	7	5	3	0
Actual	21	20	19	17	14	12	11	9	8	5	0



### 5.2 Daily Scrum Log

Tên Log	Hôm qua làm gì	Hôm nay sẽ làm gì	Khó khăn đang gặp phải
Log1			

## 6. Sprint Review

### 6.1 Nội dung Sprint Review

Những gì đã hoàn thành	Những gì đã hoàn thành	Những gì vượt phạm vi Sprint Goal	Minh chứng bằng bảng kết quả(BurnDown)
------------------------	------------------------	-----------------------------------	--

<ul style="list-style-type: none"> <li>Hoàn thành toàn bộ chức năng <b>Đăng ký</b> (<b>EM01</b>)</li> <li>Hoàn thành toàn bộ chức năng <b>Đăng nhập</b> (<b>EM02</b>)</li> <li>Hoàn thành chức năng <b>Tạo bài viết</b> (<b>EM03</b>)</li> <li>Hoàn thành chức năng <b>Xem chi tiết bài viết</b> (<b>EM04</b>)</li> <li>Hoàn thành chức năng <b>Gắn tag cho bài viết</b> (<b>EM05</b>)</li> <li>Một phần của <b>Danh sách bài viết</b> (<b>EM06</b>) được hoàn thành</li> <li>Một phần của <b>Chia sẻ bài viết</b> (<b>EM07</b>) được hoàn thành</li> </ul> <p>→ <b>Tổng task hoàn thành: 14 / 21 task</b></p>	<p>(Do thực tế Burndown cho thấy vẫn còn 7 task chưa done)</p> <ul style="list-style-type: none"> <li>Chưa hoàn tất toàn bộ chức năng <b>Danh sách bài viết</b> (<b>EM06</b>)</li> <li>Chưa hoàn tất toàn bộ chức năng <b>Chia sẻ bài viết</b> (<b>EM07</b>)</li> <li>Một số test case nâng cao chưa thực hiện</li> <li>Chưa hoàn thành tối ưu hóa UI trên mobile</li> <li>Một số lỗi UI nhỏ chưa fix</li> </ul> <p>→ <b>Lý do chưa hoàn thành:</b></p> <ul style="list-style-type: none"> <li>Các task backend/logic của EM06, EM07 phức tạp hơn dự kiến</li> <li>Một số lỗi phát sinh khiến Dev mất thời gian sửa</li> <li>Khối lượng testing thực tế lớn hơn ước tính ban đầu</li> </ul>	<p>Trong Sprint phát sinh thêm một số yêu cầu không có trong backlog ban đầu:</p> <ul style="list-style-type: none"> <li>Thêm tính năng <b>Responsive UI</b> cho mobile</li> <li>Thêm một số yêu cầu <i>UI chỉnh sửa chi tiết</i> (không có trong Sprint Goal)</li> <li>Thêm test case mới cho API share</li> <li>Phát sinh nhiệm vụ <b>bug-fix UI</b> làm chậm tiến độ chính</li> </ul> <p>→ Đây là <b>scope tăng thêm</b>, khiến một số User Story cuối sprint không hoàn thành.</p>	
--	---	--	--

## 6.2 Slide/PDF Review (nếu có)

## 7. Sprint Retrospective

### 7.1 Mô tả cảm xúc

Glad (Hài lòng)	Mad (Khó chịu / Bực mình)	Sad (Buồn / Thất vọng)
-----------------	---------------------------	------------------------

Team phối hợp tốt, trao đổi nhanh, nhiều lỗi được fix sớm.	Một số task bị kéo dài do yêu cầu thay đổi giữa chừng (scope creep).	Không hoàn thành toàn bộ Sprint Goal theo kế hoạch ban đầu.
UI/UX hoàn thiện đẹp, người dùng thử nghiệm đánh giá tốt.	Một vài lỗi phát sinh liên tục làm mất thời gian kiểm thử lại.	Một số chức năng như EM06, EM07 chưa hoàn tất do quá tải công việc.
Một số bạn chủ động nhận thêm task để hỗ trợ team.	Ước lượng Story Point chưa chính xác nên tải công việc bị dồn cuối Sprint.	Burndown Chart không đạt đường lý tưởng, tiến độ chậm ở giữa Sprint.

## 7.2 Ưu tiên cho các cải tiến quy trình

STT	Cải tiến đề xuất	Mức ưu tiên	Lý do chọn	Kết quả mong đợi
1	Rà soát lại phân chia Task trước khi vào Sprint	High	Sprint này nhiều task bị ước lượng sai, phải chỉnh giữa chừng	Giảm trễ tiến độ, Burndown mượt hơn
2	Tăng tần suất trao đổi giữa Dev – Tester	High	Có lỗi phát hiện muộn, mất thời gian sửa	Hạn chế bug tồn đọng cuối Sprint
3	Ghi nhật ký thay đổi (Change Log) rõ ràng hơn	Medium	Một số thay đổi chức năng không được ghi lại	Minh bạch, dễ kiểm tra và review
4	Chuẩn hoá quy trình viết Test Case	Medium	Test steps chưa nhất quán, khó tái kiểm	Tester làm việc nhanh hơn, ít sai sót
5	Kiểm tra sớm UI/UX trước khi Dev backend	Low	Một số UI sửa sau khi Dev xong → tốn effort	Tránh rework, tiết kiệm thời gian

## 7.3 Bài học kinh nghiệm

- Ước lượng phải sát thực tế hơn: Khi phân nhỏ User Story và hiểu rõ phạm vi, việc ước lượng Story Point và effort sẽ chính xác hơn, tránh bị trễ ở cuối Sprint.
- Luôn trao đổi sớm khi có vướng mắc: Khi một thành viên gặp block và không báo ngay, cả team bị chậm theo. Việc trao đổi sớm giúp xử lý nhanh và giữ tiến độ.
- Ưu tiên chất lượng hơn số lượng: Làm ít nhưng đúng, hạn chế commit gấp rút vì dễ gây bug phải sửa lại.
- Tester cần tham gia từ đầu Sprint: Giúp phát hiện sớm lỗi UI/logic, giảm số bug lớn ở cuối Sprint.
- Ghi lại thay đổi rõ ràng và liên tục: Giúp việc review, test lại và đổi chiểu Sprint Goal dễ dàng hơn.

- Tập trung vào Sprint Goal, tránh làm tràn phạm vi: Một số việc ngoài phạm vi làm tiêu tốn effort và ảnh hưởng tiến độ chính.