

Community App

Content Management

**Contents**

[1 Overview 3](#_Toc204870313)

[2 Scenarios 3](#_Toc204870314)

[3 Business Requirements 5](#_Toc204870315)

[4 Functional Requirements 6](#_Toc204870316)

[5 Dependencies 6](#_Toc204870317)

[6 Checklist 6](#_Toc204870318)

[7 Change and Review history 7](#_Toc204870319)

**Quick Spec – Content Management**

Author: Tahta Alfionita (tahta@slerate.com)

This document is intended to be used as part of the SLERATE spec Development and Review Process.

Confidentiality Notice

This document is an internal working document of SLERATE. It contains trade secrets and other proprietary information, which are the confidential property of SLERATE.  Neither this document nor the information contained within is to be distributed—in whole or part— outside of the SLERATE team by any means without prior authorization, including but not limited to printed, magnetic, electronic or verbal forms.

1. Overview

The Content Management feature allows Admins, Content Creators, and Managers to upload, categorize, schedule, and update various types of content (articles, resources, videos, etc.) to be accessed by community members. This feature ensures that content is organized, easy to manage, and accessible based on roles and permissions.

1. Scenarios

|  |  |
| --- | --- |
| **Scenario Name** | **Create and scheduling content** |
| Background | Connie (Content Creator) prepares a new resource. Instead of being published directly, the content must first go through Manjit (Manager) for review and approval before it becomes publicly available to Sales Professionals in the app. |
| Objective | Content Creators must be able to submit content for internal review, and Managers must have the ability to review, approve, or reject content before it is published to the platform. |
| Persona(s) | Connie – Content Creator  Manjit – Manager |
| Scenario Steps | 1. Connie logs into the SLERATE Admin Panel using her credentials. 2. From the left menu, she clicks on “Content Management” and selects “Add New Content.” 3. A content creation form appears with the following fields and fills them in:  |  |  | | --- | --- | | Field | Input | | Title | “B2B Cold Outreach Playbook – 2025 Edition” | | Content Type | PDF | | Description | “A practical template and strategy guide for B2B cold emails.” | | File Upload | [article.pdf] (2.1 MB) | | Cover Image | [cover-article.jpg] | | Publish Option | Schedule for 12/07/25 at 09:00 AM |  1. Connie clicks “Submit for Approval.” 2. The system validates input; if errors occur, inform Connie to correct before submission. 3. Upon successful submission, confirmation message appears: “Your content has been submitted for review by the Manager.” 4. Manjit receives a notification alert in the dashboard and email: “New Content Pending Approval.” 5. Manjit logs into the Admin Panel and goes to “Content Approval Queue.” 6. He clicks “Review” on the pending item and sees a content preview with the following options:    1. Approve    2. Reject (with reason)    3. Request Revisions 7. Based on the decision:    1. If Approved: System marks content as “Scheduled for publishing” and notifies Connie. Content becomes scheduled.    2. If Rejected: System marks content as “Rejected”, sends notification to Connie with manager's comments. Content does not publish.    3. If Changes Requested: System marks content as “Changes Requested”, notifies Connie with detailed feedback to revise content. 8. The content now appears in the “Scheduled Posts” tab of the dashboard. |

|  |  |
| --- | --- |
| **Scenario Name** | **Edit existing content** |
| Background | Connie needs to update previously published or drafted content. |
| Objective | Allow Content creator to safely edit content with validation and version control, handling errors proactively. |
| Persona(s) | Connie – Content Manager |
| Scenario Steps | 1. Connie logs in and navigates to the content list. 2. Searches or filters for the desired content. 3. Opens content details and clicks “Edit.” 4. Modifies fields as needed. 5. Clicks “Save” or “Submit for Approval.” 6. System validates inputs. 7. If validation fails, descriptive error messages are shown. Example for missing required fields, show customized validation errors. 8. If validation passes, changes are saved as a new version. 9. If version save fails due to backend issues, show: *“Failed to save changes. Please retry.”* 10. System confirms update success. |

|  |  |
| --- | --- |
| **Scenario Name** | **Filter and searches content** |
| Background | Connie looks for specific content by keywords, categories, or status to manage or review. |
| Objective | Provide fast, responsive filtering and searching with error feedback if search parameters are invalid. |
| Persona(s) | Connie or Content Manager tasked with content organization and review. |
| Scenario Steps | 1. Connie navigates to content management list page. 2. Inputs keyword or select filters (status, category). 3. The system validates:    1. If search input contains invalid characters or unsupported query syntax, display: *“Invalid search terms. Please revise and try again.”*    2. If no content matches criteria, show: *“No content items found matching your filters.”*    3. For server or network failures, show: *“Unable to retrieve content currently. Please try again later.”*    4. If validates is successful, the system performs search/filter and displays results. 4. Connie reviews result and select content for further action. |

1. Business Requirements

|  |  |  |
| --- | --- | --- |
|  | **Business Requirement** | **Priority** |
| 1 | Content creators can create, edit, publish, unpublish, and delete content items. | High |
| 2 | Support multiple content types (text, images, files). | High |
| 3 | Content should support categorization and tagging for easy organization. | High |
| 4 | Content editing supports text formatting and media embedding. | High |
| 5 | Workflow system for content approval and publication by authorized roles. | High |
| 6 | Scheduled publishing and unpublishing of content. | Medium |
| 7 | Content visibility controlled by access permissions. | High |
| 8 | Audit logs for all content changes and publication activities. | Medium |
| 9 | Content search and filter features to quickly locate content items. | High |
| 10 | Responsive UI supporting easy content management on different devices. | High |

1. Functional Requirements

|  |  |  |
| --- | --- | --- |
|  | **Functional Requirement** | **Business Requirement(s)** |
| 1 | Content Creator can access a content management dashboard listing all content items with status and metadata. | 1, 9 |
| 2 | Support creation of new content via forms with text editor and media upload capabilities. | 1, 4 |
| 3 | Content items can be categorized using predefined categories and tagged with keywords. | 3 |
| 4 | Admins can edit existing content; edits are tracked in log. | 1 |
| 5 | Content Creator can submit content for approval workflow; Manager can accept, request changes, or reject. | 5 |
| 6 | Publish/unpublish actions can be done manually or scheduled by date/time. | 6 |
| 7 | Content visibility is controlled by role-based permissions determining who can view or edit content. | 7 |
| 8 | Search and filter mechanisms support searching content by title, tags, categories, date, status. | 9 |
| 9 | System logs all content creation, edits, deletions, publication actions with user and timestamp. | 8 |
| 10 | UI should support responsive design for desktops, tablets, and mobile devices. | 10 |

1. Dependencies

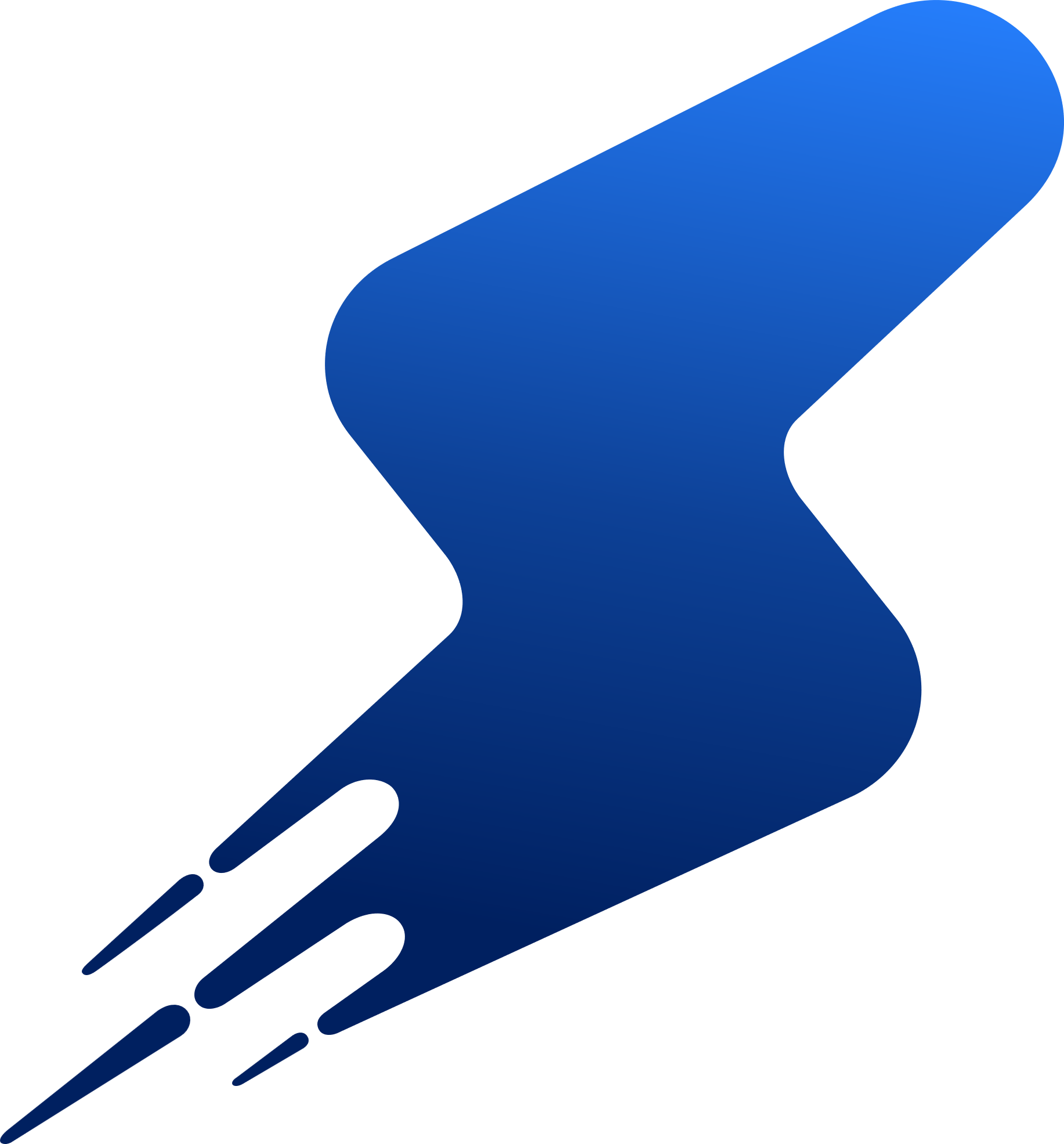
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Team/Contact** | **Type** | **Description** |
| 1 | Frontend Developer | Technical | Build UI/UX for uploading, previewing, scheduling, and organizing content |
| 2 | Backend Developer | Technical | Set up API and backend logic for storing, retrieving, and updating content |
| 3 | Content Creator Team | Operational | Responsible for uploading and managing community content |

1. Checklist

|  |  |  |
| --- | --- | --- |
|  | **Checklist Item** | **Response** |
| 1 | Has this functional spec been reviewed by Dev? | No |
| 2 | Has this functional spec been reviewed by UI? | No |
| 3 | Has this functional spec been reviewed by Manager? | No |
| 4 | CMS interface for uploading and organizing content is developed | Yes |
| 5 | Content categories and metadata (e.g., tags, access level, publish date) are defined | Yes |
| 6 | Permission-based content visibility logic is implemented | Yes |
| 7 | File storage (e.g., videos, images, docs) integrated with secure storage (e.g., AWS S3) | No |

1. Change and Review history

| **Date** | **Author, editor, or Reviewer** | **Description of change(s)/Comments on review** |
| --- | --- | --- |
| 31/7/2025 | Tahta (Author) | Create this quick specs document |
|  |  |  |
|  |  |  |



SLERATE

July 2025

Copyright © SLERATE

www.SLERATE.com