# Software Requirements Specification (SRS)

**Project:** Social Media Management Software  
**Version:** 1.0  
**Authors:** Mevin Jose,Prateek Meher, K Abhiram, K Rajeev  
**Date:** 31-08-2025  
**Status:** Model Answer

## Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Change summary | Approval |
| 1.0 | 31-08-2025 | Instructor | SRS with diagrams embedded |  |
| 1.1 | 22-10-2025 | Mevin Jose | User Stories rewritten to align with INVEST principles |  |

## Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature / Email | Date |
| Course Coordinator |  |  |  |

## Table of Contents

1. Introduction  
2. Overall description  
3. External interfaces  
4. System features (detailed)  
5. Non-functional requirements (detailed)  
6. Quality attributes & Acceptance tests  
7. UML Use-Case Diagram  
8. Requirements Traceability Matrix (RTM)

## 1. Introduction

1.1 Purpose

This document outlines the software requirements for a **Social Media Management System (SMMS)**, detailing its functional and non-functional requirements, interfaces, and verification criteria. The SMMS will provide a unified platform for businesses, influencers, and marketing teams to manage multiple social media accounts efficiently.

1.2 Scope  
The Social Media Management System (SMMS) is a compact web application designed for users to log in, link a single social media account (either simulated or through a dummy API), create and schedule posts, and access fundamental analytics such as the number of posts and scheduled items.

1.3 Audience

**Students/Developers:** Implement the project; understand functional and non-functional requirements.

**Project Evaluators/Instructors:** Assess project scope, design, and correctness.

**End Users (Demo Users):** Use the system for login, post creation, scheduling, and basic analytics.

1.4 Definitions

**SMMS (Social Media Management System):** A platform for users to log in, create, schedule, and view basic analytics for posts.

* **Post:** Text (and optional image) entry created by a user for publishing.
* **Scheduled Post:** A post stored for future publishing at a specific date/time.
* **Published Post:** A post already released (marked published in the database).
* **Analytics:** Basic statistics on post counts (total, scheduled, published).
* **Admin:** System user managing other users.
* **User:** System user creating, scheduling, and viewing posts.

## 2. Overall description

2.1 Product perspective

* This system is a standalone web app. It simulates integration with social media platforms but can be built using a local database to store posts, scheduled tasks, and analytics.

2.2 Major product functions (detailed)

* User login and registration
* Create a post (text only, optional image upload)
* Schedule posts for a future date/time
* View scheduled and published posts
* View basic analytics (total posts, scheduled posts)

2.3 User roles and characteristics (expanded)

* **Admin** – manages users
* **User** – creates and schedules posts, views analytics

2.4 Operating environment

**Hardware Environment:**

* Any standard PC or laptop with at least 4GB RAM and 10GB free disk space.
* Optional: Can also run on a local server (XAMPP, WAMP, or Node.js environment).

**Software Environment:**

* Operating System: Windows, Linux, or macOS.
* Web Browser: Chrome, Firefox, or Edge (for accessing the web interface).
* Backend: PHP with MySQL (XAMPP/WAMP) *or* Python Django with SQLite *or* Node.js with Express and SQLite/MySQL.
* Frontend: HTML, CSS, JavaScript (basic UI).

**Network Requirements:**

* Runs locally (no internet required if API integration is not used).
* If extended with real social media APIs, requires internet access.

2.5 Constraints

* Posts are stored locally in the database (not published to real platforms, unless optional APIs are used).
* Basic analytics only (counts, not real engagement data).

## 3. External interface requirements

3.1 User interfaces

* **Primary UI:** Web-based dashboard (HTML, CSS, JS) accessible from desktop or mobile browser.
* **Features:**
  + Login/Register page with simple forms.
  + Dashboard with menu for “Create Post”, “Scheduled Posts”, “Published Posts”, and “Analytics”.
  + Post creation form with fields for text, optional image upload, and schedule date/time.
  + Analytics view showing total posts, scheduled posts, and published posts.
* **Accessibility:**
  + High-contrast mode for readability.
  + Basic keyboard navigation.

3.2 Hardware interfaces

* Standard user device (PC, laptop, or smartphone).
* Keyboard and mouse or touch input for post creation.
* Optional: File upload (for images).

3.3 Software interfaces

* **Database:** MySQL/SQLite for storing users, posts, and schedules.
* **Auth System:** Local authentication (username/password).
* **Optional APIs:** Can be extended with real social media APIs (Facebook, Twitter/X, Instagram) for actual publishing.
* **File Handling:** Local storage for uploaded images.

3.4 Communications

* Localhost access for development.
* HTTPS/TLS 1.2+ if deployed online.
* Internal communication between modules via REST endpoints (if Node.js/Django used).
* Notification messages (email or in-app) for scheduled/published posts (optional extension).

## 4. System features (detailed)

Each requirement below includes acceptance criteria and a reference test case. IDs follow SMMS-F-###.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req ID   |  | | --- | |  | | Requirement   |  | | --- | |  | | Type | Priority | Source/  Stakeholder | Acceptance Criteria/  Test case ref | Comments/  Dependencies |
| SMMS-F-001 | As a new user, I want to register using my email and password so that I can create an account and access the platform. | Authentication | High | User/Security | AC-SMMS-F-001: User enters valid info → account is created → redirected to login page. Test: TC-Auth-01 | Requires database connection |
| SMMS-F-002 | As a registered user, I want to log in using my credentials so that I can access my dashboard securely. | Authentication | High | User/Security | AC-SMMS-F-002: Valid login → dashboard loads; invalid → error message.  Test: TC-Auth-02 | Depends on F-001 |
| SMMS-F-003 | As a system user, I want my password stored securely so that my personal data remains protected | Security | High | Developer/Security | AC-SMMS-F-003: Database contains only hashed passwords.  Test: TC-Sec-01 | Requires password hashing library |
| SMMS-F-004 | As an Admin, I want to manage different roles (Admin/User) so that permissions are properly enforced. | Authorization | Medium | Admin/Instructor | AC-SMMS-F-004: Admin can see/manage user list; user cannot access admin features. Test: TC-RBAC-01 | Requires F-002 |
| SMMS-F-005 | As a user, I want to create a post with text content so that I can share updates easily. | Functional | High | User | AC-SMMS-F-005: Post saved → visible in dashboard. Test: TC-Post-01 | Needs logged in session |
| SMMS-F-006 | As a user, I want to attach an image to my post so that my post looks more engaging. | Functional | Medium | User | AC-SMMS-F-006: Image uploads successfully and appears with post. Test: TC-Post-02 | Requires file storage |
| SMMS-F-007 | As a user, I want to save my post as a draft so that I can edit or publish it later. | Functional | Medium | User | AC-SMMS-F-007: Draft saved → visible under “Drafts” tab. Test: TC-Post-03 | Depends on F-005 |
| SMMS-F-008 | As a user, I want to schedule a post for a specific time so that it publishes automatically. | Scheduling | High | User | AC-SMMS-F-008: Scheduled post displays correct date/time.  Test: TC-Sched-01 | Needs DB timestamp field |
| SMMS-F-009 | As a user, I want my scheduled post to be auto-published at the chosen time so that I don’t need to do it manually. | Scheduling | High | User | AC-SMMS-F-009: Post automatically moves to “Published” list.  Test: TC-Sched-02 | Requires background scheduler |
| SMMS-F-010 | As a user, I want to view all my scheduled posts so that I can manage future posts effectively. | Functional | Medium | User | AC-SMMS-F-010: Dashboard lists all scheduled posts.  Test: TC-View-01 | Depends on F-008 |
| SMMS-F-011 | As a user, I want to view my published posts so that I can confirm what has been posted. | Functional | Medium | User | AC-SMMS-F-011: Dashboard lists all published posts with timestamps.Test: TC-View-02 | Depends on F-009 |
| SMMS-F-012 | As a user, I want to edit or delete my scheduled posts before they are published so that I can make corrections easily. | Functional | High | User | AC-SMMS-F-012: User edits/deletes post → update reflected immediately. Test: TC-Post-04 | Depends on F-008 |
| SMMS-F-013 | As a user, I want to see analytics showing my total, scheduled, and published posts so that I can track my activity. | Analytics | Medium | User/Instructor | AC-SMMS-F-013: Analytics page shows correct counts. Test: TC-Anal-01 | Requires F-005, F-008, F-009 |
| SMMS-F-014 | As an Admin, I want to view and deactivate user accounts so that I can maintain proper system access. | Admin | Medium | Admin | AC-SMMS-F-014: Admin can deactivate users; confirmation message shown. Test: TC-Admin-01 | Requires F-004 |
| SMMS-F-015 | As a user, I want to receive success or failure notifications when I schedule or publish a post so that I know whether my action worked. | Feedback | Medium | User | AC-SMMS-F-015: Notifications appear after each action.  Test: TC-Notif-01 | Requires UI messaging |

## 5. Non-functional requirements (detailed)

NFRs below are measurable and tied to test plans. IDs SMMS-NF-###.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement | Category | Priority | Acceptance criteria / Measurement |
| SMMS-NF-001 | The system shall respond to user actions (login, post creation) within 2 seconds under normal load. | Performance | High | 90th percentile ≤ 5s in production-like test. Test: TC-Perf-01 |
| SMMS-NF-002 | The system shall support up to 50 concurrent users. | Reliability | Medium | Uptime reports show ≥99.9% per month. Test: Ops reports. |
| SMMS-NF-003 | The system UI shall be mobile-responsive and accessible via standard browsers. | Usability | Medium | PCI-DSS audit checklist pass. Test: TC-Sec-01 |
| SMMS-NF-004 | The system shall allow storage of at least 1,000 posts in the database. | Scalability | Medium | Automated archival and retrieval verified. Test: TC-OPS-01 |
| SMMS-NF-005 | The system shall allow manual backup of the database. | Maintainability | Low | Accessibility audit pass. Test: TC-UX-01 |

## 5.1. Security

## 5.1.1 Security Objectives

* **Protect User Accounts** – Ensure that user credentials and personal information are safeguarded against unauthorized access.
* **Maintain Data Confidentiality and Integrity** – Posts, schedules, and analytics data must remain accurate and protected from tampering or leakage.
* **Ensure Secure Access Control** – Only authenticated users should be able to create or schedule posts, and admin-only features must remain restricted.
* **Provide Safe System Usage** – The system should defend against common web security threats such as SQL injection, XSS, and brute-force login attempts.

## 5.1.2 Security Requirements

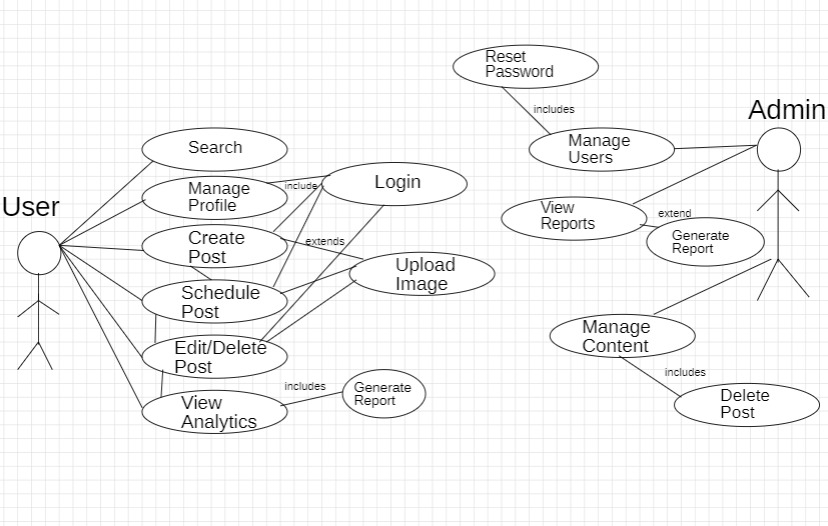
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement (shall...) | Type | Priority | Acceptance criteria / Test case ref |
| SMMS-SR-001 | The system shall store all user passwords in hashed format (e.g., bcrypt/SHA-256) and never in plain text. | Security | High | DB inspection shows no plain text passwords. Test: TC-Sec-01 |
| SMMS-SR-002 | The system shall enforce session timeouts after 15 minutes of inactivity. | Security | Medium | User automatically logged out after timeout. Test: TC-Sec-02 |
| SMMS-SR-003 | The system shall validate all inputs to prevent SQL injection and XSS attacks. | Security | High | Security tests with malicious inputs fail safely. Test: TC-Sec-03 |
| SMMS-SR-004 | The system shall restrict access so that only authenticated users can create/schedule posts, and only Admin can manage users. | Security | High | Unauthorized access attempts denied. Test: TC-Sec-04 |
| SMMS-SR-005 | If deployed online, the system shall enforce HTTPS/TLS 1.2+ for all communication. | Security | High | SSL/TLS certificate active, all requests via HTTPS. Test: TC-Sec-05 |

## 6. Quality attributes & Acceptance tests

* **Exit criteria for acceptance:**
  + All high-priority functional requirements implemented and verified.
  + No critical non-functional failures (system performance, usability).
  + All security requirements satisfied (password hashing, session timeout, restricted access).
  + RTM shows that all requirements are linked to at least one test case and verified.
* **Acceptance test suites:**
  + **Authentication Tests:** User registration, login, password hashing, session timeout.
  + **Authorization Tests:** Role-based access (Admin/User).
  + **Post Management Tests:** Create, draft, schedule, publish, edit, delete posts.
  + **Analytics Tests:** View post counts (total, scheduled, published).
  + **Admin Tests:** Manage users.
  + **Performance Tests:** Response time ≤ 2 seconds, support for 50 concurrent users.
  + **Security Tests:** SQL injection prevention, password encryption, HTTPS enforcement.
  + **Usability Tests:** Dashboard navigation, responsive UI.

## 7. System models and diagrams

## 7.1 UML Use-Case diagram



## 8. Requirements Traceability Matrix (RTM)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement short | Section ref / Design Spec | Module | Test case(s) | Status (N/P/A) | Comments |
| SMMS-F-001 | User registration | 4.1 / DS-Auth-01 | Auth Module | TC-Auth-01 | N | Requires DB |
| SMMS-F-0002 | User login | 4.1 / DS-Auth-02 | Auth Module | TC-Auth-02 | N | Depends on F-001 |
| SMMS-F-0003 | Password hashing | 4.1 / DS-Auth-03 | Security Layer | TC-Sec-01 | N | Requires hashing lib |
| SMMS-F-0004 | Role-based access | 4.1 / DS-RBAC-01 | RBAC Module | TC-RBAC-01 | N | Admin/User roles |
| SMMS-F-0005 | Create post (text) | 4.2 / DS-Post-01 | Post Module | TC-Post-01 | N | Basic CRUD |
| SMMS-F-0006 | Upload image | 4.2 / DS-Post-02 | Post Module | TC-Post-02 | N | Optional |
| SMMS-F-0007 | Save draft | 4.2 / DS-Post-03 | Post Module | TC-Post-03 | N | Draft status |
| SMMS-F-0008 | Schedule post | 4.3 / DS-Sched-01 | Scheduler | TC-Sched-01 | N | Needs Timestamp |
| SMMS-F-0009 | Auto-publish post | 4.3 / DS-Sched-02 | Scheduler | TC-Sched-02 | N | Requires cron/background |
| SMMS-F-0010 | View scheduled posts | 4.3 / DS-Sched-03 | Post Module | TC-View-01 | N | List scheduled |
| SMMS-F-0011 | View published posts | 4.3 / DS-Sched-04 | Post Module | TC-View-02 | N | List published |
| SMMS-F-0012 | Edit/delete scheduled posts | 4.3 / DS-Sched-05 | Post Module | TC-Post-04 | N | Editable before publish |
| SMMS-F-0013 | Analytics (counts) | 4.4 / DS-Analy-01 | Analytics Module | TC-Analy-01 | N | Simple counts |
| SMMS-F-0014 | Admin manage users | 4.1 / DS-Admin-01 | Admin Module | TC-Admin-01 | N | View/manage users |
| SMMS-F-0015 | Notifications | 4.2 / DS-Notif-01 | Notification Module | TC-Notif-01 | N | Success/failure alerts |
| SMMS-NF-001 | Response ≤2s | 5.0 / Perf-01 | System | TC-Perf-01 | N | Load testing |
| SMMS-NF-002 | Support 50 users | 5.0 / Rel-01 | System | TC-Load-01 | N | Concurrent test |
| SMMS-NF-003 | Responsive UI | 5.0 / UI-01 | UI Layer | TC-UI-01 | N | Browser test |
| SMMS-NF-004 | 1000 posts storage | 5.0 / DB-01 | Database | TC-DB-01 | N | Stress test |
| SMMS-NF-005 | Manual backup | 5.0 / Maint-01 | Admin Module | TC-Backup-01 | N | DB export |
| SMMS-SR-001 | Password hashing | 5.1 / Sec-01 | Security Layer | TC-Sec-01 | N | Secure storage |
| SMMS-SR-002 | Session timeout | 5.1 / Sec-02 | Session Manager | TC-Sec-02 | N | 15-min auto logout |
| SMMS-SR-003 | Input validation | 5.1 / Sec-03 | Security Layer | TC-Sec-03 | N | Prevent SQLi/XSS |
| SMMS-SR-004 | Restricted access | 5.1 / Sec-04 | Auth & RBAC | TC-Sec-04 | N | Only auth users |