

**Course: Software Construction & Development**

**Submitted to: Maam Aroosa**

**Submitted by: Umer Hayat**

**Project Name: Social Media Management Tool**

**Social Media Management Tool**

# Possible requirements

**Functional Requirements**

## Sign Up

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.01 | The system shall allow new users to sign up by submitting a form with the following details:   * Name * Email * Password | High |
| FR.02 | The system shall allow users to sign up using their Google account. | High |
| FR.03 | The system shall allow users to sign up using their Facebook account. | High |

## Login/Logout

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.04 | The system shall allow registered users to log in using:   * Email * Password | High |
| FR.05 | The system shall allow registered users to log in through Google. | High |
| FR.06 | The system shall allow registered users to log in through Facebook. | High |
| FR.07 | The system shall allow users to reset their account password via email. | High |
| FR.08 | The system shall allow users to log out of their accounts. | High |

## Post Creation

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.09 | The system shall allow users to create social media posts with captions and media (images/videos). | High |
| FR.10 | The system shall provide AI-based suggestions for:   * Post captions * Relevant hashtags | High |
| FR.11 | The system shall provide basic image editing tools (e.g., resizing, filters). | Medium |
| FR.12 | The system shall allow users to save posts as drafts for later use. | High |

## Post Scheduling

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.13 | The system shall allow users to schedule posts for specific dates and times. | High |
| FR.14 | The system shall display scheduled posts in a calendar interface. | High |
| FR.15 | The system shall allow users to edit or reschedule a previously scheduled post. | High |
| FR.16 | The system shall send notifications/reminders for upcoming scheduled posts. | Medium |

## Content Planner

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.17 | The system shall allow users to organize weekly content plans using a drag-and-drop calendar. | High |
| FR.18 | The system shall allow users to tag posts by categories (e.g., campaign, theme). | Medium |
| FR.19 | The system shall provide AI-based recommendations for the best times to post. | High |

## Analytics Dashboard

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.20 | The system shall display performance metrics for each post, including:   * Likes * Shares * Comments * Engagement rate | High |
| FR.21 | The system shall allow users to compare performance metrics across platforms. | High |
| FR.22 | The system shall generate weekly or monthly analytics reports in PDF format. | Medium |

## Collaboration Features

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.23 | The system shall allow team members to collaborate on post creation and scheduling. | High |
| FR.24 | The system shall provide an approval workflow for content drafts before publishing. | Medium |
| FR.25 | The system shall allow team members to leave comments on drafts. | Medium |

## Third-Party Integrations

|  |  |  |
| --- | --- | --- |
| **SR. No** | **Description** | **Priority** |
| FR.26 | The system shall integrate with major social media platforms (Facebook, Instagram, Twitter, LinkedIn). | High |
| FR.27 | The system shall integrate with cloud storage services like Google Drive and Dropbox. | Medium |

**Non-Functional Requirements**

**Security**

* The system shall ensure all passwords and sensitive data, such as user credentials and media files, are stored in encrypted form and protected with secure sockets (SSL/TLS).
* The system shall enforce role-based access control (RBAC) and secure login mechanisms (e.g., OAuth, MFA) to restrict unauthorized access.
* The system shall log all failed login attempts, notify users of suspicious activities, and apply regular security patches to prevent vulnerabilities.
* The system shall include a robust backup and recovery plan to safeguard data during breaches or failures.

**Usability**

* The system shall have a user-friendly, responsive interface with intuitive navigation and properly labeled components across all devices.
* The system shall provide clear tooltips, concise error messages, and visually distinct analytics reports for enhanced user experience.
* The system shall include help resources such as FAQs and tutorials to assist users in resolving issues.

**Performance**

* The system shall support up to **10,000 concurrent users** and ensure **99.9% uptime** for all core functionalities.
* The system shall respond to user actions and load dashboards, content calendars, and analytics data within **3 seconds** under normal conditions.
* The system shall recover from system failures or API outages within **15 minutes** to minimize disruptions.

**Flexibility**

* The system shall be scalable to integrate new features and support additional social media platforms without requiring major architectural changes.
* The system shall adapt to multiple screen sizes, browsers, and devices seamlessly.

**Responsiveness**

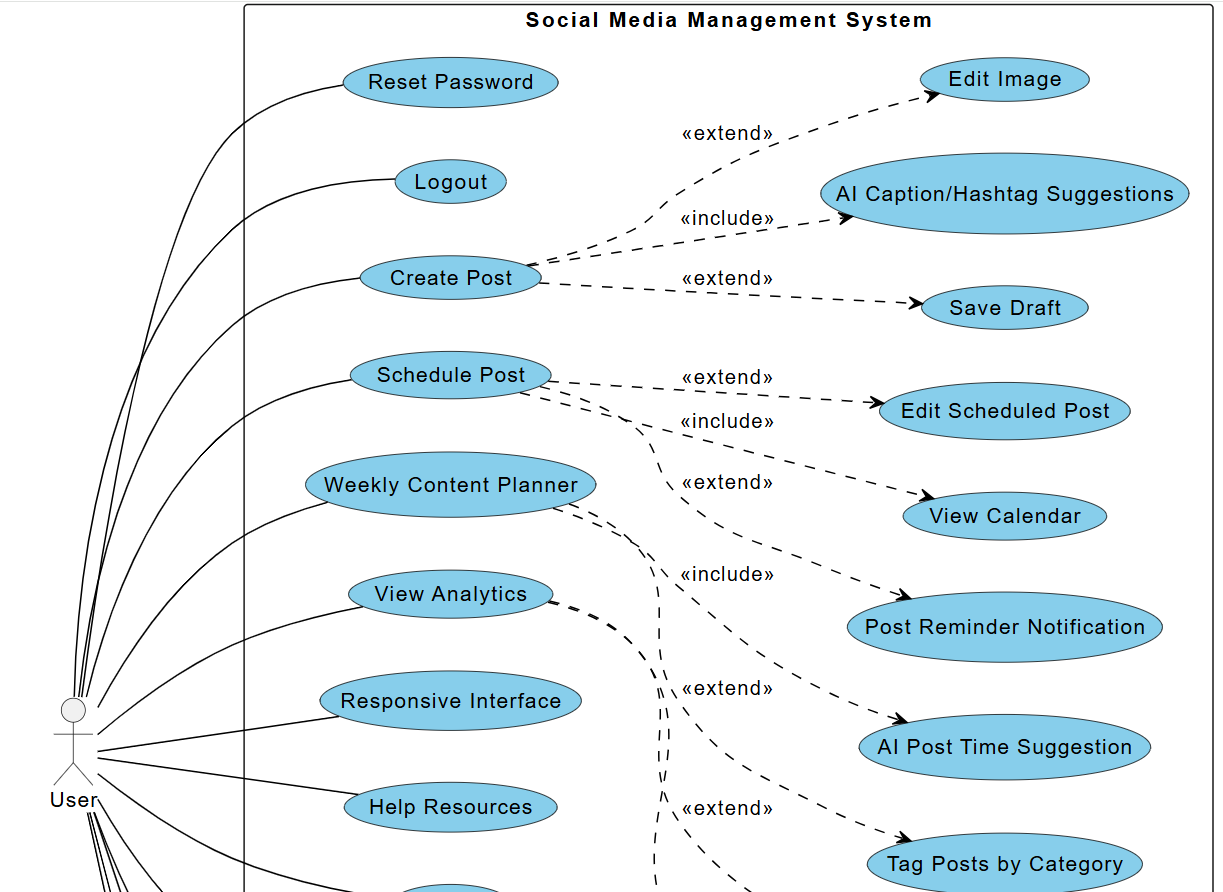
* The system shall ensure consistent functionality and content layout across all browsers (Chrome, Firefox, Safari, Edge) and devices.
* The system shall maintain usability and content integrity across various screen resolutions and orientations.

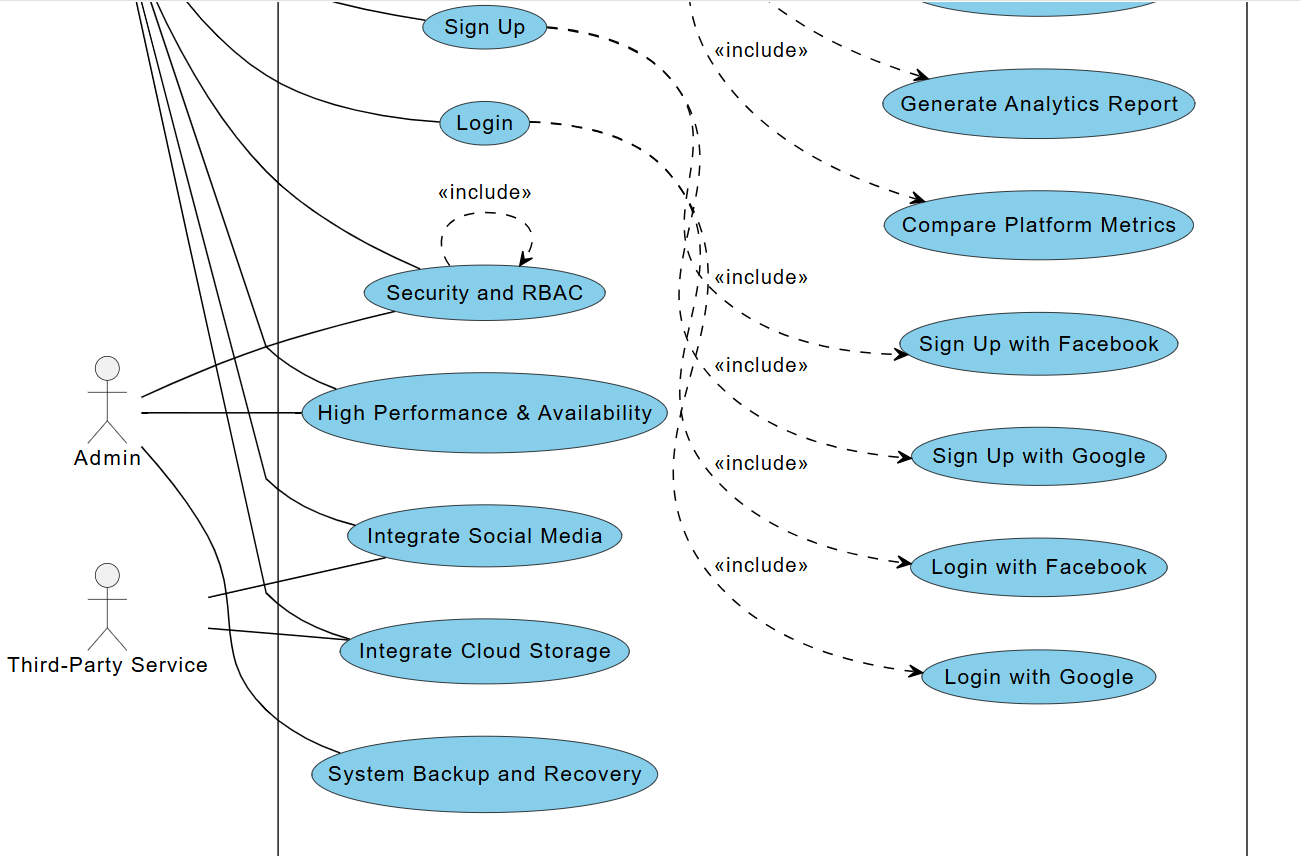
**Availability**

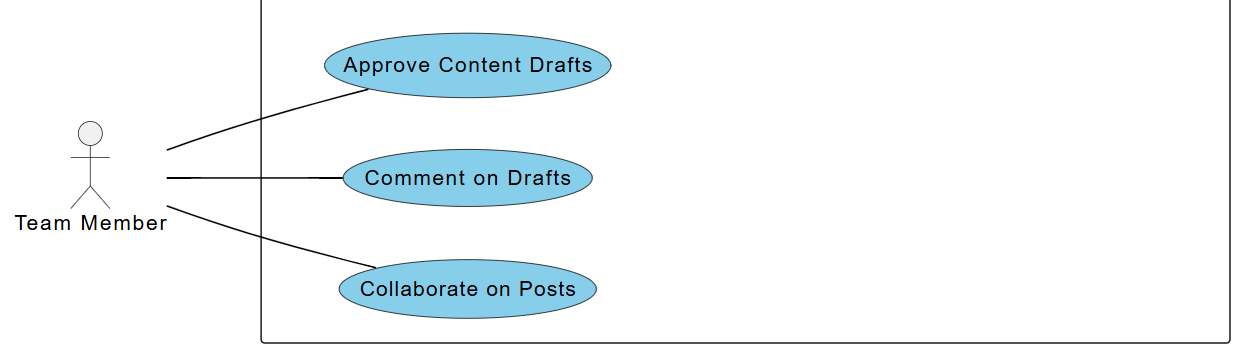
* The system shall ensure **24/7 availability** with a **99.9% uptime guarantee**, except during planned maintenance or updates.
* The system shall notify users in advance of downtime and include automatic recovery mechanisms to minimize service disruptions.

**UML Diagrams**

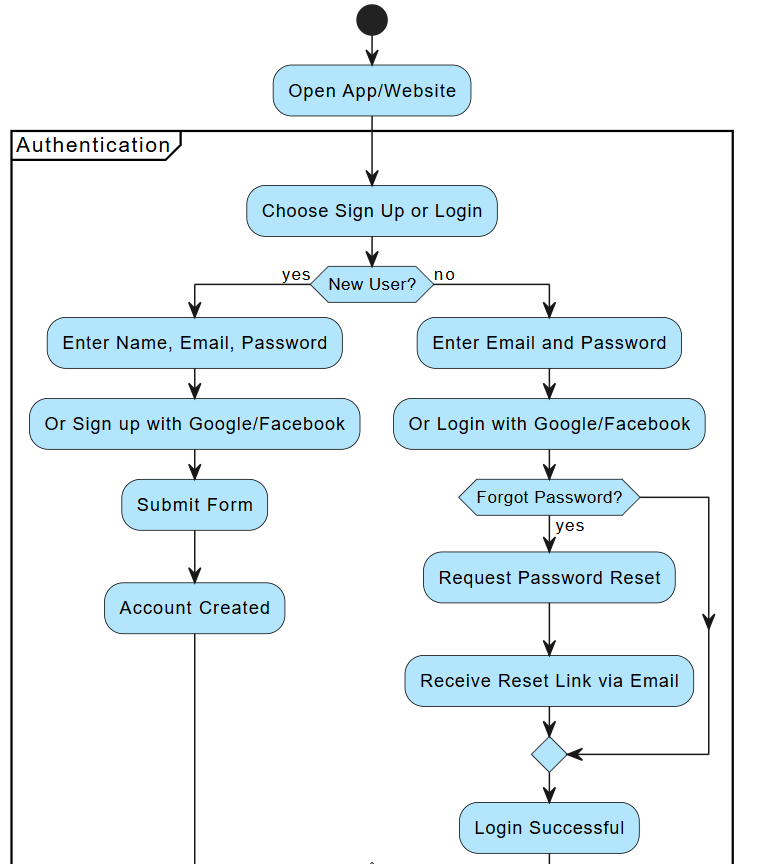
**Use Case Diagram**

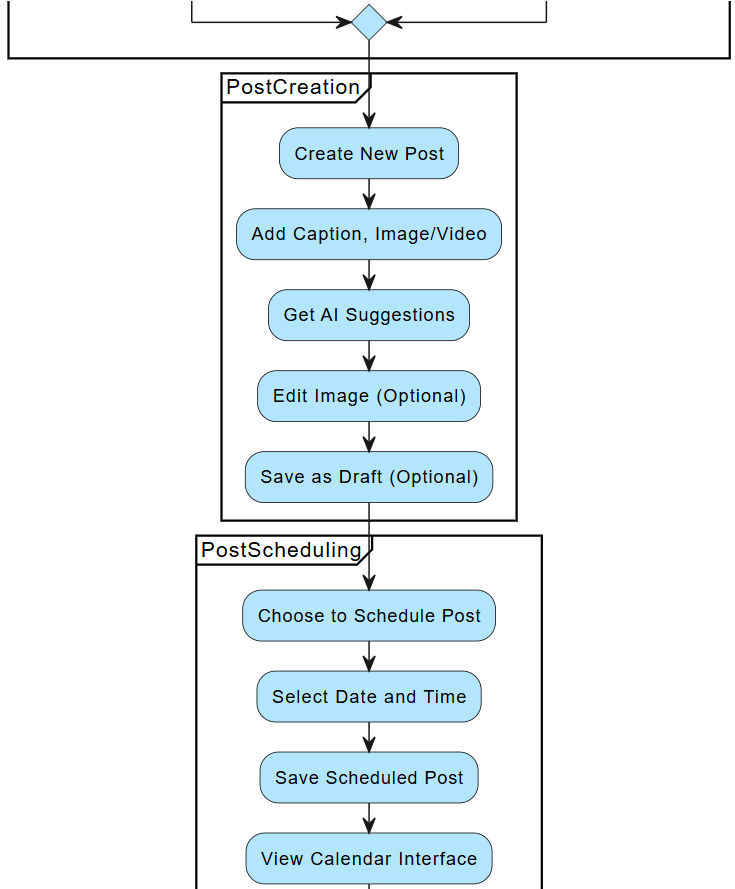


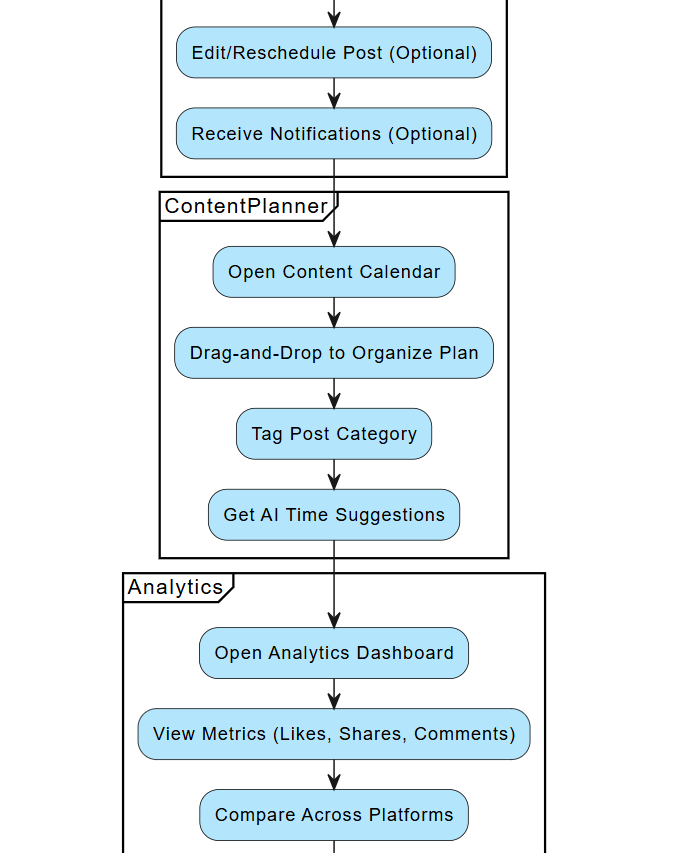


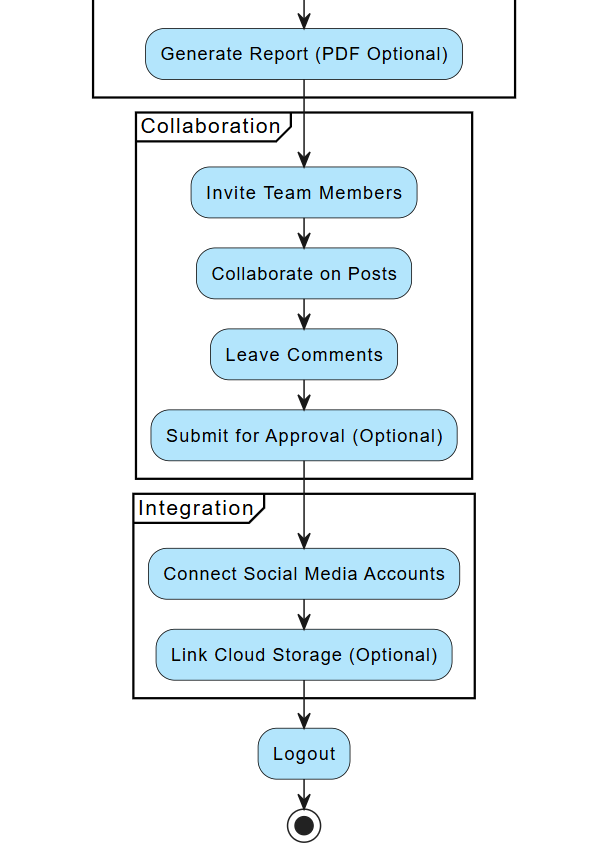


# Activity Diagram

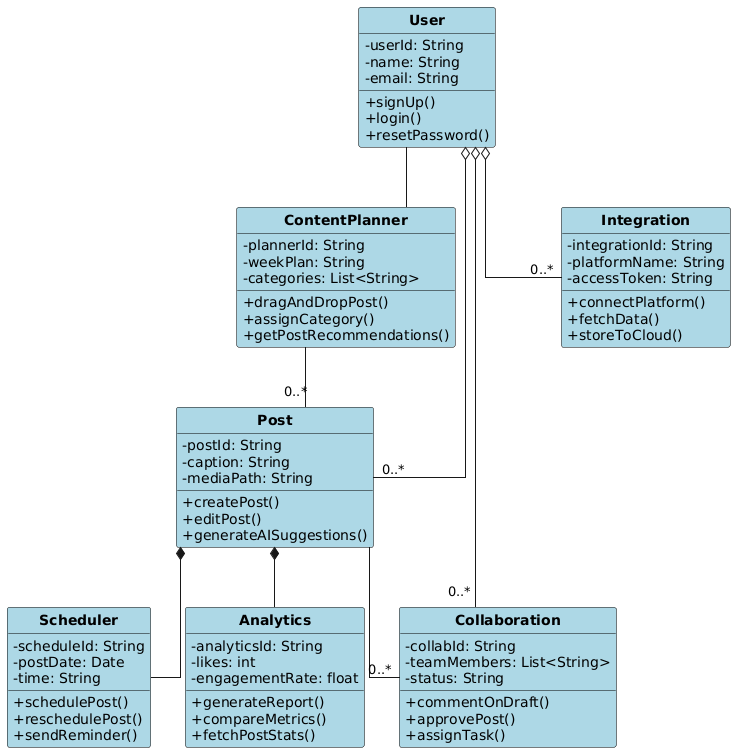




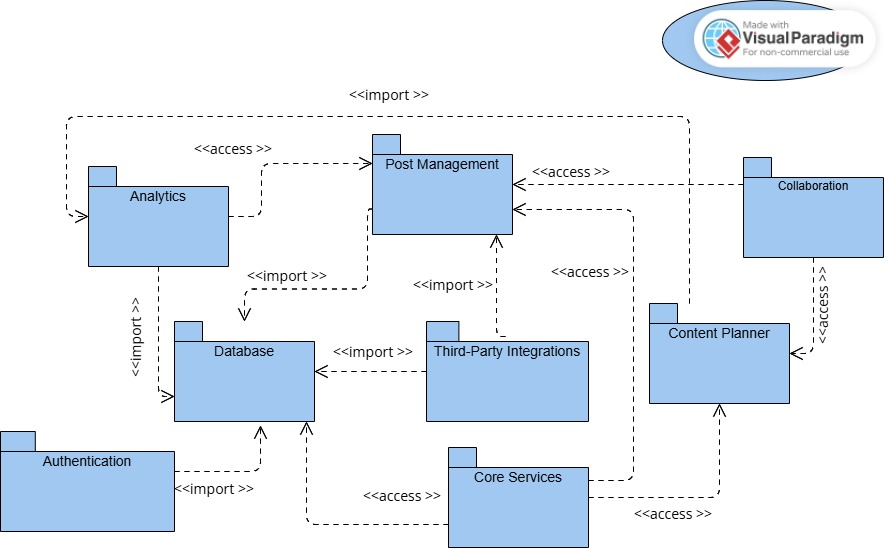




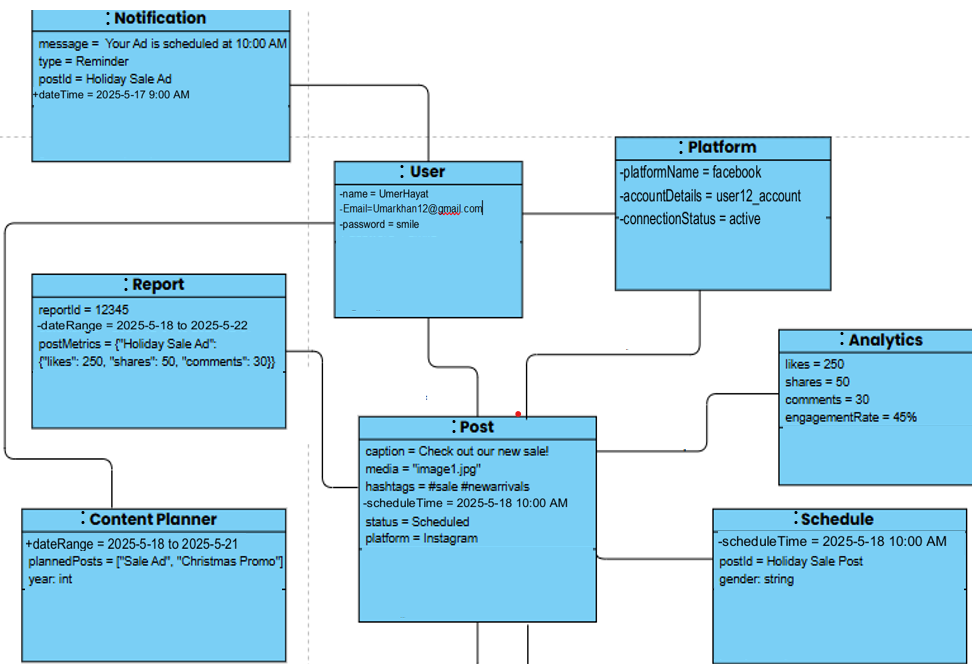
# Class Diagram

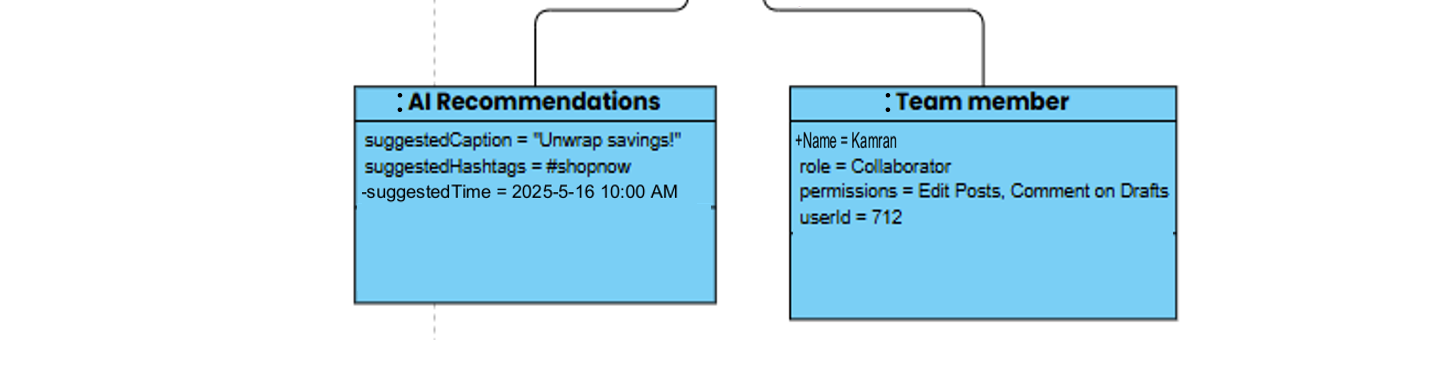


# Package Diagram

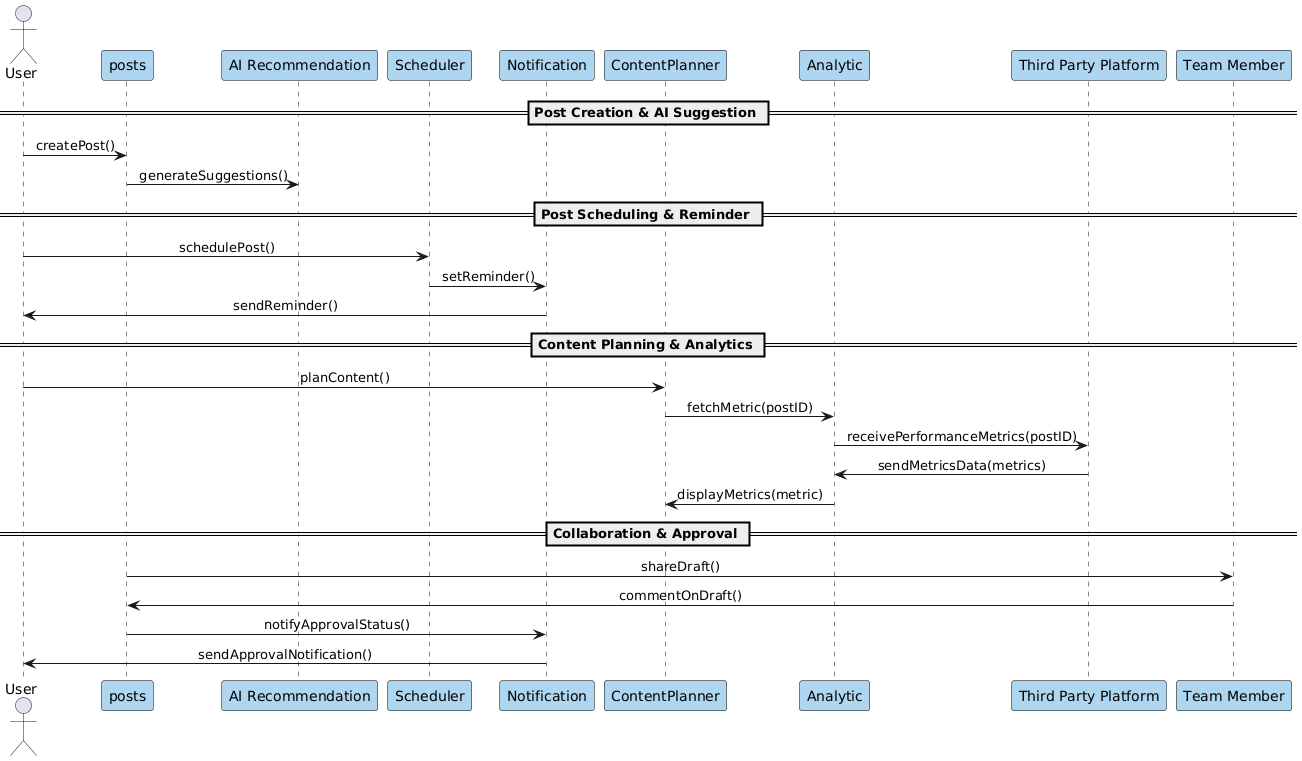


# Object Diagram

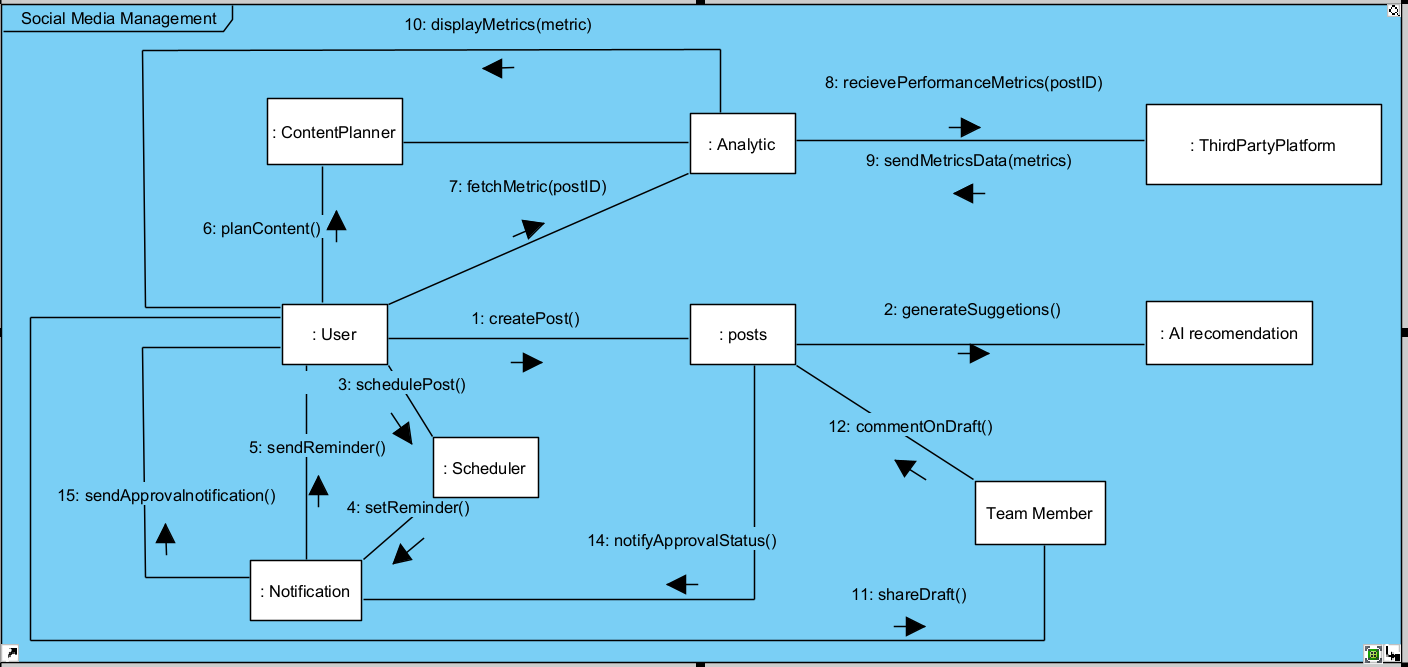




# Sequence Diagram

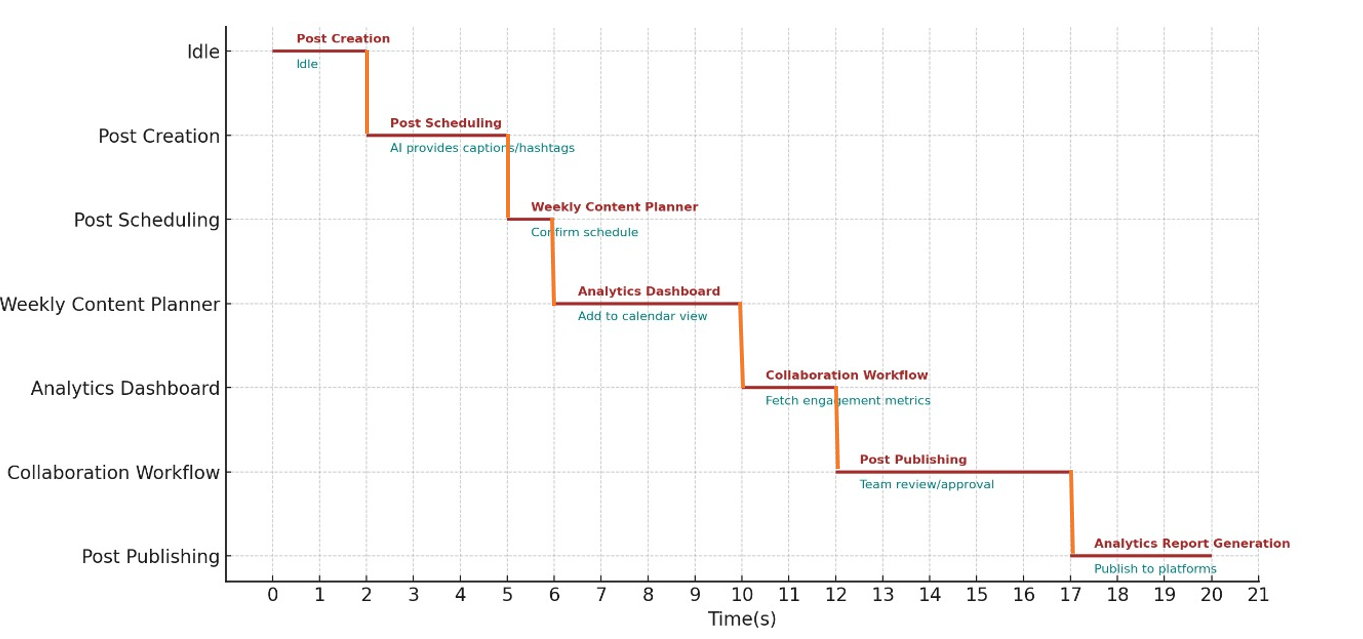


# Communication Diagram



**:**

# Timing Diagram



# Bad code to clean code

## Bad Code

import java.util.ArrayList;

import java.util.List;

class Post {

    private String t;   //meaningless variables

    private String c;

    private String sT;

    public Post(String t, String c) {   //overlapping

        this.t = t;

        this.c = c;

        this.sT = "Not Scheduled";

    }

    public void setST(String st) {

        this.st = st;

    }

}

public class SocialMediaManager {

    private List<Post> posts = new ArrayList<>();

    public void manage() {

        add("Post 1", "Hello world!");

        add("Post 2", "Java is fun!");

        add("Post 3", "Follow us for updates!");

        schedulePost(0, "2025-5-18 10:00");

        schedulePost(1, "2025-5-22 11:00");

        schedulePost(2, "2025-5-30 12:00");

        publish(0);

        publish(1);

        publish(2);

        showPost();

    }

    public void addPost(String t, String c) {

        posts.add(new Post(t, c));

        System.out.println("Added: " + t);

    }

    public void schedulePost(int i, String time) {

        if (isValidi(i)) {

            posts.get(i).setScheduleTime(time);

            System.out.println("Scheduled: " + posts.get(i));

        } else {

            System.out.println("Invalid post i.");

        }

    }

    public void publish(int i) {

        if (isValidi(i)) {

            System.out.println("Publishing: " + posts.get(i));

        } else {

            System.out.println("Invalid post i.");

        }

    }

    public void showPost() {

        System.out.println("\nAll Posts:");

        for (int i = 0; i < posts.size(); i++) {

            System.out.println((i + 1) + ". " + posts.get(i));

        }

    }

    private boolean isValidi(int i) {

        return i >= 0 && i < posts.size();

    }

    public static void main(String[] args) {

        new SocialMediaManager().manage();

    }

}

## Clean Code

import java.util.ArrayList;

import java.util.List;

class Post {

    private String title;

    private String content;

    private String scheduleTime;

    public Post(String title, String content) {

        this.title = title;

        this.content = content;

        this.scheduleTime = "Not Scheduled";

    }

    public void setScheduleTime(String scheduleTime) {

        this.scheduleTime = scheduleTime;

    }

    @Override

    public String toString() {

        return "Post[Title: " + title + ", Content: " + content + ", Scheduled: " + scheduleTime + "]";

    }

}

public class SocialMediaManager {

    private List<Post> posts = new ArrayList<>();

    public void manage() {

        addPost("Post 1", "Hello world!");

        addPost("Post 2", "Java is fun!");

        addPost("Post 3", "Follow us for updates!");

        schedulePost(0, "2025-5-18 10:00");

        schedulePost(1, "2025-5-22 11:00");

        schedulePost(2, "2025-5-30 12:00");

        publishPost(0);

        publishPost(1);

        publishPost(2);

        displayAllPosts();

    }

    public void addPost(String title, String content) {

        posts.add(new Post(title, content));

        System.out.println("Added: " + title);

    }

    public void schedulePost(int index, String time) {

        if (isValidIndex(index)) {

            posts.get(index).setScheduleTime(time);

            System.out.println("Scheduled: " + posts.get(index));

        } else {

            System.out.println("Invalid post index.");

        }

    }

    public void publishPost(int index) {

        if (isValidIndex(index)) {

            System.out.println("Publishing: " + posts.get(index));

        } else {

            System.out.println("Invalid post index.");

        }

    }

    public void displayAllPosts() {

        System.out.println("\nAll Posts:");

        for (int i = 0; i < posts.size(); i++) {

            System.out.println((i + 1) + ". " + posts.get(i));

        }

    }

    private boolean isValidIndex(int index) {

        return index >= 0 && index < posts.size();

    }

    public static void main(String[] args) {

        new SocialMediaManager().manage();

    }

}

# Comments in the cleaned code

/\*

 \* LEGAL COMMENT:

 \* This software is provided "as is" without warranty of any kind, either express or implied  \*/

import java.util.ArrayList;

import java.util.List;

// Represents a social media post.

class Post {

    private String title;

    private String content;

    private String scheduleTime;

    public Post(String title, String content) {

        this.title = title;

        this.content = content;

        this.scheduleTime = "Not Scheduled";

    }

    /\* WARNING:

     Ensure that the schedule time is not in the past to avoid logic errors.\

    \*/

    public void setScheduleTime(String scheduleTime) {

        this.scheduleTime = scheduleTime;

    }

}

public class SocialMediaManager {

    private List<Post> posts = new ArrayList<>();

    /\*

     INFORMATIVE COMMENT:

     Entry point for managing social media posts. This method encapsulates operations like creating,

     scheduling, and publishing posts.

     \*/

    public void manage() {

        // EXPLANATION OF INTENT:

        // Add sample posts for demonstration purposes.

        addPost("Post 1", "Hello world!");

        addPost("Post 2", "Java is fun!");

        addPost("Post 3", "Follow us for updates!");

        schedulePost(0, "2025-5-18 10:00");

        schedulePost(1, "2025-5-22 11:00");

        schedulePost(2, "2025-5-30 12:00");

        publishPost(0);

        publishPost(1);

        publishPost(2);

        displayAllPosts();

    }

    public void addPost(String title, String content) {

        // CLARIFICATION:

        // Each post is represented as an object of the Post class to encapsulate related data.

        posts.add(new Post(title, content));

        System.out.println("Added: " + title);

    }

    public void schedulePost(int index, String time) {

        if (isValidIndex(index)) {

            posts.get(index).setScheduleTime(time);

            System.out.println("Scheduled: " + posts.get(index));

        } else {

            System.out.println("Invalid post index.");

        }

    }

    public void publishPost(int index) {

        if (isValidIndex(index)) {

            System.out.println("Publishing: " + posts.get(index));

        } else {

            System.out.println("Invalid post index.");

        }

    }

 /\*TODO COMMENT:

   Enhance this method to filtering and sorting posts by different criteria.

  \*/

    public void displayAllPosts() {

        System.out.println("\nAll Posts:");

        for (int i = 0; i < posts.size(); i++) {

            System.out.println((i + 1) + ". " + posts.get(i));

        }

    }

    private boolean isValidIndex(int index) {

        return index >= 0 && index < posts.size();

    }

    public static void main(String[] args) {

        new SocialMediaManager().manage();

    }

}

## OutPut

