

Product Requirements Document (PRD)

CreatorSync: Cross-Platform Content Creator Management Suite

Document Information

- **Document Title:** CreatorSync Product Requirements Document
- **Version:** 1.0
- **Last Updated:** March 23, 2025
- **Status:** Draft

Table of Contents

1. Introduction
2. Product Overview
3. User Personas
4. User Stories
5. Functional Requirements
6. Non-Functional Requirements
7. User Interface Requirements
8. System Architecture
9. Data Requirements
10. Integration Requirements
11. Security Requirements
12. Performance Requirements
13. Compliance Requirements
14. Assumptions and Constraints
15. Success Metrics
16. Future Considerations
17. Appendices

Introduction

Purpose

This Product Requirements Document (PRD) outlines the specifications and requirements for CreatorSync, a comprehensive SaaS platform designed to help content creators manage their presence across TikTok, Instagram Reels, and X (Twitter). The document serves as the definitive source of information for the development team, stakeholders, and future reference.

Scope

CreatorSync will address the key pain points identified in our research on content creators' challenges across multiple platforms. The platform will include tools for content management, analytics, monetization tracking, audience engagement, and strategic growth.

Document Conventions

- **Must Have:** Essential features required for launch
- **Should Have:** Important features that provide significant value but aren't critical for launch
- **Could Have:** Desirable features that will be implemented if time and resources permit
- **Won't Have:** Features explicitly excluded from the current scope

Intended Audience

- Development Team

- Product Management
- UX/UI Designers
- QA Team
- Marketing Team
- Executive Stakeholders

References

- Content Creator Pain Points Analysis (March 2025)
- Market Analysis Report (February 2025)
- Competitive Analysis of Creator Tools (January 2025)
- Platform API Documentation (TikTok, Instagram, X)

Product Overview

Product Vision

CreatorSync will be the definitive all-in-one platform that empowers content creators to efficiently manage, optimize, and monetize their presence across multiple social platforms, saving time while maximizing audience growth and revenue.

Product Goals

1. Simplify cross-platform content management for creators
2. Provide actionable analytics to optimize content strategy
3. Streamline monetization tracking and optimization
4. Enhance audience engagement and community management
5. Support strategic growth across multiple platforms

Target Market

Content creators with 10,000+ followers across TikTok, Instagram Reels, and X, segmented into: - Emerging Creators (10K-50K followers) - Established Creators (50K-500K followers) - Professional Creators (500K+ followers)

Key Differentiators

1. **Creator-First Approach:** Designed specifically for content creators' unique needs
2. **Cross-Platform Integration:** Seamless management across all three platforms
3. **AI-Powered Optimization:** Data-driven recommendations for content strategy
4. **Comprehensive Business Tools:** Complete suite for professional creators
5. **Unified Analytics:** Cross-platform performance insights

Product Positioning

CreatorSync positions itself as the professional operating system for serious content creators, bridging the gap between basic scheduling tools and enterprise-level marketing suites with a solution specifically tailored to creator needs.

User Personas

Persona 1: Emily - Emerging Creator

- **Demographics:** 24 years old, fashion and lifestyle creator
- **Platforms:** TikTok (35K followers), Instagram (25K followers), X (15K followers)
- **Technical Proficiency:** Moderate
- **Content Creation Process:** Creates content alone using smartphone and basic editing apps

- **Pain Points:**
 - Struggles to maintain consistent posting schedule across platforms
 - Limited understanding of which content performs best
 - Beginning to receive brand deals but has no system to manage them
 - Spends too much time on manual platform management
- **Goals:**
 - Grow followers to qualify for more monetization programs
 - Increase efficiency to post more consistently
 - Start earning sustainable income from content

Persona 2: Marcus - Established Creator

- **Demographics:** 32 years old, tech reviewer and educator
- **Platforms:** TikTok (120K followers), Instagram (180K followers), X (90K followers)
- **Technical Proficiency:** High
- **Content Creation Process:** Works with a small team (video editor and assistant)
- **Pain Points:**
 - Difficulty coordinating content calendar with team
 - Managing multiple brand partnerships simultaneously
 - Tracking revenue across affiliate programs, platform monetization, and sponsorships
 - Analyzing performance across platforms to optimize strategy
- **Goals:**
 - Scale content production without sacrificing quality
 - Optimize monetization strategy across platforms
 - Build more predictable revenue streams

Persona 3: Sophia - Professional Creator

- **Demographics:** 29 years old, fitness and wellness entrepreneur
- **Platforms:** TikTok (750K followers), Instagram (1.2M followers), X (350K followers)
- **Technical Proficiency:** Moderate
- **Content Creation Process:** Works with a team of 5 (content manager, videographer, editor, community manager, business manager)
- **Pain Points:**
 - Complex workflow management across team members
 - Enterprise-level needs for analytics and reporting
 - Managing multiple revenue streams and business relationships
 - Maintaining brand consistency across platforms
- **Goals:**
 - Optimize team workflow and collaboration
 - Make data-driven decisions about content strategy
 - Maximize ROI on content production
 - Expand business opportunities beyond platform monetization

User Stories

Content Management

1. As a creator, I want to schedule content across multiple platforms from a single interface so that I can save time and maintain a consistent posting schedule.
 - **Acceptance Criteria:**
 - Can upload content once and schedule for multiple platforms
 - Can customize captions, hashtags, and other metadata per platform
 - Receives notifications for scheduled posts
 - Can view calendar of scheduled content across all platforms

2. As a creator, I want to receive platform-specific optimization recommendations so that my content performs better on each platform.
 - **Acceptance Criteria:**
 - Receives suggestions for optimal posting times
 - Gets recommendations for hashtags and keywords
 - Receives format optimization tips (aspect ratio, duration, etc.)
 - Gets alerts about trending topics relevant to my niche
3. As a creator, I want to repurpose content across platforms efficiently so that I can maximize the value of each piece of content.
 - **Acceptance Criteria:**
 - Can adapt content format for different platforms (aspect ratios, durations)
 - Can remove platform watermarks
 - Can extract clips from longer content
 - Can add platform-specific elements (text, stickers, etc.)

Analytics

4. As a creator, I want to view unified analytics across all platforms so that I can understand my overall performance.
 - **Acceptance Criteria:**
 - Can view key metrics from all platforms in a single dashboard
 - Can compare performance across platforms
 - Can track growth trends over time
 - Can export reports for brand partnerships
5. As a creator, I want to understand which content performs best so that I can optimize my content strategy.
 - **Acceptance Criteria:**
 - Can view performance metrics by content type, topic, and format
 - Receives AI-powered insights about high-performing content
 - Can compare similar content across different platforms
 - Can track engagement patterns over time
6. As a creator, I want to understand my audience demographics across platforms so that I can create more targeted content.
 - **Acceptance Criteria:**
 - Can view unified audience demographics
 - Can identify audience overlap between platforms
 - Can segment audience by engagement level
 - Can track audience growth and retention

Monetization

7. As a creator, I want to track all my revenue streams in one place so that I can understand my overall business performance.
 - **Acceptance Criteria:**
 - Can connect platform monetization programs
 - Can track brand deals and sponsorships
 - Can monitor affiliate marketing performance
 - Can view revenue forecasts and historical data
8. As a creator, I want to manage brand partnerships efficiently so that I can maintain professional relationships and deliver on time.
 - **Acceptance Criteria:**
 - Can track partnership status and deliverables
 - Can store contracts and communication history
 - Receives deadline reminders
 - Can generate performance reports for partners

9. As a creator, I want to identify new monetization opportunities so that I can diversify my revenue streams.

- **Acceptance Criteria:**

- Receives recommendations for relevant affiliate programs
- Gets alerts about platform monetization eligibility
- Can view monetization performance by content type
- Can compare revenue potential across different opportunities

Community Engagement

10. As a creator, I want to manage comments and messages across platforms so that I can engage with my audience efficiently.

- **Acceptance Criteria:**

- Can view and respond to comments from all platforms in one interface
- Can filter and prioritize comments based on engagement potential
- Can save and categorize common responses
- Can track response rates and times

11. As a creator, I want to identify and engage with high-value followers so that I can build stronger community relationships.

- **Acceptance Criteria:**

- Can identify most engaged followers across platforms
- Can tag and categorize followers
- Can track interaction history with specific followers
- Can create targeted engagement strategies

12. As a creator, I want to analyze sentiment and topics in audience feedback so that I can better understand audience preferences.

- **Acceptance Criteria:**

- Can view sentiment analysis of comments and messages
- Can identify common topics and questions
- Can track sentiment trends over time
- Can compare audience feedback across platforms

Growth Strategy

13. As a creator, I want to receive strategic growth recommendations so that I can expand my audience effectively.

- **Acceptance Criteria:**

- Receives platform-specific growth recommendations
- Gets content strategy suggestions based on performance data
- Can view competitive benchmarking
- Can set and track growth goals

14. As a creator, I want to test different content strategies so that I can optimize my approach.

- **Acceptance Criteria:**

- Can create and track A/B tests
- Can compare performance of different content types
- Receives analysis of test results
- Can implement winning strategies across platforms

15. As a creator, I want to stay informed about platform changes and trends so that I can adapt my strategy accordingly.

- **Acceptance Criteria:**

- Receives alerts about algorithm updates
- Gets notifications about new features and tools
- Can view trending topics in my niche
- Receives industry news and insights

Team Collaboration

16. As a creator team, we want to collaborate on content planning and creation so that we can work efficiently together.
 - **Acceptance Criteria:**
 - Can assign tasks and track progress
 - Can share content assets and feedback
 - Can approve content before publishing
 - Can communicate within the platform
17. As a creator team, we want to define roles and permissions so that team members can access appropriate features.
 - **Acceptance Criteria:**
 - Can create custom roles with specific permissions
 - Can assign team members to roles
 - Can limit access to sensitive information
 - Can track team member activity

Functional Requirements

1. Unified Dashboard Module

1.1 Platform Connections

- **FR1.1.1:** System must support secure OAuth connection to TikTok, Instagram, and X accounts
- **FR1.1.2:** System must detect and alert users about connection issues
- **FR1.1.3:** System must support multiple accounts per platform
- **FR1.1.4:** System must refresh authentication tokens automatically when needed

1.2 Dashboard Views

- **FR1.2.1:** System must provide customizable dashboard layouts
- **FR1.2.2:** System must display real-time platform status and notifications
- **FR1.2.3:** System must show key performance metrics from all connected platforms
- **FR1.2.4:** System must allow filtering dashboard by date ranges and platforms

1.3 Notifications

- **FR1.3.1:** System must aggregate notifications from all platforms
- **FR1.3.2:** System must allow customization of notification preferences
- **FR1.3.3:** System must support in-app, email, and mobile push notifications
- **FR1.3.4:** System must provide notification center for historical alerts

2. Content Studio Module

2.1 Content Calendar

- **FR2.1.1:** System must provide a unified calendar view of content across platforms
- **FR2.1.2:** System must support drag-and-drop scheduling
- **FR2.1.3:** System must allow recurring content scheduling
- **FR2.1.4:** System must support calendar filtering by platform, content type, and status

2.2 Content Creation and Editing

- **FR2.2.1:** System must support uploading videos, images, and text content
- **FR2.2.2:** System must provide basic editing tools (trimming, cropping, filters)
- **FR2.2.3:** System must support platform-specific content formatting
- **FR2.2.4:** System must allow adding captions, hashtags, and other metadata

2.3 Content Scheduling

- **FR2.3.1:** System must support scheduling content for specific dates and times
- **FR2.3.2:** System must recommend optimal posting times based on audience data
- **FR2.3.3:** System must support queue-based posting
- **FR2.3.4:** System must allow bulk scheduling of content

2.4 Content Optimization

- **FR2.4.1:** System must provide platform-specific optimization recommendations
- **FR2.4.2:** System must suggest relevant hashtags and keywords
- **FR2.4.3:** System must alert users about content that may violate platform policies
- **FR2.4.4:** System must provide readability and engagement predictions

2.5 Content Library

- **FR2.5.1:** System must maintain a searchable library of all content
- **FR2.5.2:** System must support tagging and categorization of content
- **FR2.5.3:** System must track usage and performance of content
- **FR2.5.4:** System must support content templates and saved elements

3. Analytics Command Center Module

3.1 Cross-Platform Analytics

- **FR3.1.1:** System must aggregate key metrics across all platforms
- **FR3.1.2:** System must normalize metrics for cross-platform comparison
- **FR3.1.3:** System must visualize performance trends over time
- **FR3.1.4:** System must support custom date ranges for analysis

3.2 Content Performance Analytics

- **FR3.2.1:** System must track performance metrics for all content
- **FR3.2.2:** System must provide comparative analysis between content pieces
- **FR3.2.3:** System must identify patterns in high-performing content
- **FR3.2.4:** System must correlate content attributes with performance

3.3 Audience Analytics

- **FR3.3.1:** System must aggregate audience demographics across platforms
- **FR3.3.2:** System must identify audience overlap between platforms
- **FR3.3.3:** System must track audience growth and retention
- **FR3.3.4:** System must segment audience by engagement level

3.4 Custom Reports

- **FR3.4.1:** System must support creation of custom report templates
- **FR3.4.2:** System must allow scheduling of automated reports
- **FR3.4.3:** System must support export in multiple formats (PDF, CSV, etc.)
- **FR3.4.4:** System must allow sharing reports with team members and partners

3.5 Competitive Analysis

- **FR3.5.1:** System must allow tracking of competitor accounts
- **FR3.5.2:** System must provide benchmarking against industry averages
- **FR3.5.3:** System must identify trending content in the user's niche
- **FR3.5.4:** System must alert users about significant competitor activities

4. Monetization Hub Module

4.1 Revenue Tracking

- **FR4.1.1:** System must track revenue from platform monetization programs
- **FR4.1.2:** System must support manual entry of brand deal and sponsorship revenue
- **FR4.1.3:** System must integrate with major affiliate marketing platforms
- **FR4.1.4:** System must visualize revenue trends over time

4.2 Brand Deal Management

- **FR4.2.1:** System must provide a CRM for brand relationships
- **FR4.2.2:** System must track deal status, deliverables, and deadlines
- **FR4.2.3:** System must store contracts and communication history
- **FR4.2.4:** System must generate performance reports for brand partners

4.3 Affiliate Marketing

- **FR4.3.1:** System must track affiliate link performance
- **FR4.3.2:** System must generate and manage trackable links
- **FR4.3.3:** System must recommend relevant affiliate programs
- **FR4.3.4:** System must analyze content-to-conversion performance

4.4 Monetization Strategy

- **FR4.4.1:** System must provide revenue forecasting
- **FR4.4.2:** System must recommend monetization opportunities based on audience
- **FR4.4.3:** System must analyze revenue per content type
- **FR4.4.4:** System must suggest pricing for brand partnerships based on market data

4.5 Financial Management

- **FR4.5.1:** System must track expenses related to content creation
- **FR4.5.2:** System must calculate profit margins and ROI
- **FR4.5.3:** System must generate financial reports for accounting
- **FR4.5.4:** System must support tax documentation preparation

5. Community Engagement Suite Module

5.1 Unified Inbox

- **FR5.1.1:** System must aggregate comments and messages from all platforms
- **FR5.1.2:** System must support filtering and searching of communications
- **FR5.1.3:** System must allow batch responses to similar comments
- **FR5.1.4:** System must track response rates and times

5.2 Automated Responses

- **FR5.2.1:** System must suggest responses based on message content
- **FR5.2.2:** System must support saved response templates
- **FR5.2.3:** System must allow creation of automated response rules
- **FR5.2.4:** System must support scheduling of delayed responses

5.3 Audience Relationship Management

- **FR5.3.1:** System must identify and track high-value followers
- **FR5.3.2:** System must maintain interaction history with followers
- **FR5.3.3:** System must support tagging and categorization of followers

- **FR5.3.4:** System must alert users about engagement opportunities

5.4 Sentiment Analysis

- **FR5.4.1:** System must analyze sentiment in comments and messages
- **FR5.4.2:** System must identify common topics and questions
- **FR5.4.3:** System must track sentiment trends over time
- **FR5.4.4:** System must alert users about negative sentiment spikes

5.5 Community Building

- **FR5.5.1:** System must suggest engagement strategies based on audience data
- **FR5.5.2:** System must track community growth and engagement metrics
- **FR5.5.3:** System must support community challenges and campaigns
- **FR5.5.4:** System must identify potential community advocates

6. Growth Accelerator Module

6.1 Strategic Recommendations

- **FR6.1.1:** System must provide platform-specific growth recommendations
- **FR6.1.2:** System must suggest content strategies based on performance data
- **FR6.1.3:** System must recommend collaboration opportunities
- **FR6.1.4:** System must identify underperforming areas for improvement

6.2 A/B Testing

- **FR6.2.1:** System must support creation and tracking of A/B tests
- **FR6.2.2:** System must analyze test results and provide insights
- **FR6.2.3:** System must recommend implementation of winning strategies
- **FR6.2.4:** System must maintain history of tests and results

6.3 Trend Monitoring

- **FR6.3.1:** System must track trending topics in the user's niche
- **FR6.3.2:** System must alert users about relevant trends
- **FR6.3.3:** System must analyze trend lifecycle and recommend timing
- **FR6.3.4:** System must suggest content ideas based on trends

6.4 Platform Intelligence

- **FR6.4.1:** System must track platform algorithm changes
- **FR6.4.2:** System must alert users about new platform features
- **FR6.4.3:** System must provide guidance on adapting to platform changes
- **FR6.4.4:** System must analyze impact of platform changes on performance

6.5 Goal Setting and Tracking

- **FR6.5.1:** System must support setting growth and performance goals
- **FR6.5.2:** System must track progress toward goals
- **FR6.5.3:** System must provide recommendations to achieve goals
- **FR6.5.4:** System must adjust forecasts based on actual performance

7. Team Collaboration Module

7.1 Task Management

- **FR7.1.1:** System must support creation and assignment of tasks
- **FR7.1.2:** System must track task status and deadlines
- **FR7.1.3:** System must send notifications about task updates
- **FR7.1.4:** System must provide task dashboards and reports

7.2 Content Workflow

- **FR7.2.1:** System must support customizable content approval workflows
- **FR7.2.2:** System must track content through production stages
- **FR7.2.3:** System must allow feedback and revisions on content
- **FR7.2.4:** System must maintain version history of content

7.3 Asset Management

- **FR7.3.1:** System must provide centralized storage for creative assets
- **FR7.3.2:** System must support organization and tagging of assets
- **FR7.3.3:** System must track asset usage and rights
- **FR7.3.4:** System must support sharing assets with team members

7.4 Communication

- **FR7.4.1:** System must provide in-app messaging between team members
- **FR7.4.2:** System must support commenting on specific content items
- **FR7.4.3:** System must maintain communication history
- **FR7.4.4:** System must allow sharing of reports and analytics

7.5 Roles and Permissions

- **FR7.5.1:** System must support creation of custom roles
- **FR7.5.2:** System must allow granular permission settings
- **FR7.5.3:** System must track user activity and changes
- **FR7.5.4:** System must support single sign-on for team members

8. Account and Settings Module

8.1 User Management

- **FR8.1.1:** System must support user registration and authentication
- **FR8.1.2:** System must allow profile management
- **FR8.1.3:** System must support password reset and account recovery
- **FR8.1.4:** System must provide activity logs for security

8.2 Subscription Management

- **FR8.2.1:** System must support multiple subscription tiers
- **FR8.2.2:** System must handle billing and payment processing
- **FR8.2.3:** System must allow upgrading/downgrading subscription
- **FR8.2.4:** System must provide subscription usage analytics

8.3 Preferences

- **FR8.3.1:** System must allow customization of notification preferences
- **FR8.3.2:** System must support personalization of dashboard views
- **FR8.3.3:** System must remember user-specific settings

- **FR8.3.4:** System must support language and timezone preferences

8.4 Integration Management

- **FR8.4.1:** System must allow management of platform connections
- **FR8.4.2:** System must support integration with third-party tools
- **FR8.4.3:** System must provide API access for custom integrations
- **FR8.4.4:** System must track integration status and health

Non-Functional Requirements

1. Performance Requirements

1.1 Response Time

- **NFR1.1.1:** Web application must load initial dashboard in under 2 seconds
- **NFR1.1.2:** API responses must complete within 500ms for 95% of requests
- **NFR1.1.3:** Data visualization components must render in under 1 second
- **NFR1.1.4:** Content uploads must process at a minimum of 10MB per second

1.2 Scalability

- **NFR1.2.1:** System must support at least 100,000 concurrent users
- **NFR1.2.2:** System must handle at least 1,000 API requests per second
- **NFR1.2.3:** Database must support at least 10TB of content and analytics data
- **NFR1.2.4:** System must scale horizontally to accommodate growth

1.3 Availability

- **NFR1.3.1:** System must maintain 99.9% uptime (less than 8.76 hours of downtime per year)
- **NFR1.3.2:** Planned maintenance must be scheduled during off-peak hours
- **NFR1.3.3:** System must implement redundancy for critical components
- **NFR1.3.4:** System must recover from failures within 5 minutes

2. Security Requirements

2.1 Authentication and Authorization

- **NFR2.1.1:** System must implement multi-factor authentication
- **NFR2.1.2:** System must enforce strong password policies
- **NFR2.1.3:** System must implement role-based access control
- **NFR2.1.4:** System must support OAuth 2.0 for platform integrations

2.2 Data Protection

- **NFR2.2.1:** All data must be encrypted in transit using TLS 1.3 or higher
- **NFR2.2.2:** Sensitive data must be encrypted at rest using AES-256
- **NFR2.2.3:** System must implement data loss prevention measures
- **NFR2.2.4:** System must support secure deletion of user data

2.3 Compliance

- **NFR2.3.1:** System must comply with GDPR requirements
- **NFR2.3.2:** System must comply with CCPA requirements
- **NFR2.3.3:** System must implement privacy by design principles
- **NFR2.3.4:** System must maintain audit logs for compliance purposes

2.4 Vulnerability Management

- **NFR2.4.1:** System must undergo regular security assessments
- **NFR2.4.2:** Critical vulnerabilities must be patched within 24 hours
- **NFR2.4.3:** System must implement intrusion detection and prevention
- **NFR2.4.4:** System must have a responsible disclosure policy

3. Usability Requirements

3.1 User Interface

- **NFR3.1.1:** UI must be responsive and support all major browsers
- **NFR3.1.2:** UI must be accessible according to WCAG 2.1 AA standards
- **NFR3.1.3:** UI must support dark mode and light mode
- **NFR3.1.4:** UI must maintain consistent design language across all modules

3.2 User Experience

- **NFR3.2.1:** System must provide onboarding tutorials for new users
- **NFR3.2.2:** System must implement progressive disclosure of complex features
- **NFR3.2.3:** System must provide contextual help throughout the interface
- **NFR3.2.4:** System must support keyboard shortcuts for common actions

3.3 Localization

- **NFR3.3.1:** System must support multiple languages (initially English, Spanish, French)
- **NFR3.3.2:** System must handle different date, time, and number formats
- **NFR3.3.3:** System must support right-to-left languages in future updates
- **NFR3.3.4:** System must allow user-specific language preferences

4. Reliability Requirements

4.1 Error Handling

- **NFR4.1.1:** System must provide clear error messages to users
- **NFR4.1.2:** System must log detailed error information for troubleshooting
- **NFR4.1.3:** System must handle API rate limiting from platforms gracefully
- **NFR4.1.4:** System must implement circuit breakers for external dependencies

4.2 Data Integrity

- **NFR4.2.1:** System must maintain ACID compliance for database transactions
- **NFR4.2.2:** System must implement data validation for all inputs
- **NFR4.2.3:** System must maintain audit trails for data changes
- **NFR4.2.4:** System must perform regular data integrity checks

4.3 Backup and Recovery

- **NFR4.3.1:** System must perform automated backups at least daily
- **NFR4.3.2:** System must retain backups for at least 30 days
- **NFR4.3.3:** System must support point-in-time recovery
- **NFR4.3.4:** System must test recovery procedures regularly

5. Compatibility Requirements

5.1 Browser Compatibility

- **NFR5.1.1:** System must support latest versions of Chrome, Firefox, Safari, and Edge

- **NFR5.1.2:** System must support at least the previous two major versions of each browser
- **NFR5.1.3:** System must gracefully degrade on older browsers
- **NFR5.1.4:** System must be responsive on tablets and mobile devices

5.2 Platform Compatibility

- **NFR5.2.1:** System must adapt to TikTok API changes within 48 hours
- **NFR5.2.2:** System must adapt to Instagram API changes within 48 hours
- **NFR5.2.3:** System must adapt to X API changes within 48 hours
- **NFR5.2.4:** System must maintain compatibility with platform policy changes

5.3 Integration Compatibility

- **NFR5.3.1:** System must provide REST API for third-party integrations
- **NFR5.3.2:** System must support webhook notifications for events
- **NFR5.3.3:** System must implement standard authentication for API access
- **NFR5.3.4:** System must provide comprehensive API documentation

User Interface Requirements

1. Design System

1.1 Visual Design

- **UIR1.1.1:** System must implement a consistent color scheme across all modules
- **UIR1.1.2:** System must use a modern, clean typography system
- **UIR1.1.3:** System must implement consistent spacing and layout grids
- **UIR1.1.4:** System must use platform-specific visual cues (colors, icons) for clarity

1.2 Component Library

- **UIR1.2.1:** System must use a consistent component library
- **UIR1.2.2:** Components must be accessible and support keyboard navigation
- **UIR1.2.3:** Components must provide appropriate feedback for user actions
- **UIR1.2.4:** Components must be responsive to different screen sizes

1.3 Iconography and Imagery

- **UIR1.3.1:** System must use consistent iconography throughout
- **UIR1.3.2:** Icons must be recognizable and follow platform conventions where appropriate
- **UIR1.3.3:** System must optimize images for fast loading
- **UIR1.3.4:** System must support alt text for all images

2. Layout and Navigation

2.1 Dashboard Layout

- **UIR2.1.1:** Dashboard must use a modular, customizable layout
- **UIR2.1.2:** Dashboard must prioritize key metrics and notifications
- **UIR2.1.3:** Dashboard must support different views based on user role
- **UIR2.1.4:** Dashboard must be optimized for quick scanning of information

2.2 Navigation Structure

- **UIR2.2.1:** System must implement a consistent global navigation
- **UIR2.2.2:** Navigation must clearly indicate current location
- **UIR2.2.3:** Navigation must provide shortcuts to frequently used features
- **UIR2.2.4:** Navigation must adapt to different screen sizes

2.3 Information Architecture

- **UIR2.3.1:** Content must be organized in a logical hierarchy
- **UIR2.3.2:** Related information must be grouped together
- **UIR2.3.3:** System must implement progressive disclosure for complex features
- **UIR2.3.4:** System must provide clear paths for common user journeys

3. Interactive Elements

3.1 Forms and Inputs

- **UIR3.1.1:** Forms must validate input in real-time
- **UIR3.1.2:** Forms must provide clear error messages
- **UIR3.1.3:** Forms must support autofill where appropriate
- **UIR3.1.4:** Forms must be optimized for completion efficiency

3.2 Data Visualization

- **UIR3.2.1:** Charts and graphs must be clear and accurately represent data
- **UIR3.2.2:** Visualizations must be interactive where appropriate
- **UIR3.2.3:** Visualizations must be accessible with alternative text representations
- **UIR3.2.4:** Visualizations must adapt to different screen sizes

3.3 Feedback and Notifications

- **UIR3.3.1:** System must provide immediate feedback for user actions
- **UIR3.3.2:** Notifications must be non-intrusive but noticeable
- **UIR3.3.3:** System must allow customization of notification preferences
- **UIR3.3.4:** System must aggregate notifications to prevent overload

System Architecture

1. High-Level Architecture

1.1 Architecture Overview

- **AR1.1.1:** System must implement a microservices architecture
- **AR1.1.2:** System must use containerization for deployment
- **AR1.1.3:** System must implement API gateway for service access
- **AR1.1.4:** System must use event-driven architecture for real-time features

1.2 Frontend Architecture

- **AR1.2.1:** Frontend must use a component-based framework
- **AR1.2.2:** Frontend must implement state management for complex interactions
- **AR1.2.3:** Frontend must use responsive design principles
- **AR1.2.4:** Frontend must implement progressive web app capabilities

1.3 Backend Architecture

- **AR1.3.1:** Backend must implement RESTful API design
- **AR1.3.2:** Backend must use domain-driven design principles
- **AR1.3.3:** Backend must implement caching for performance optimization
- **AR1.3.4:** Backend must use asynchronous processing for long-running tasks

2. Infrastructure

2.1 Cloud Infrastructure

- **AR2.1.1:** System must be deployed on cloud infrastructure
- **AR2.1.2:** System must implement auto-scaling for demand fluctuations
- **AR2.1.3:** System must use content delivery network for static assets
- **AR2.1.4:** System must implement multi-region deployment for disaster recovery

2.2 Database Architecture

- **AR2.2.1:** System must use relational database for transactional data
- **AR2.2.2:** System must use NoSQL database for content and analytics data
- **AR2.2.3:** System must implement database sharding for scalability
- **AR2.2.4:** System must use read replicas for query performance

2.3 Caching Strategy

- **AR2.3.1:** System must implement application-level caching
- **AR2.3.2:** System must use distributed cache for shared data
- **AR2.3.3:** System must implement cache invalidation strategies
- **AR2.3.4:** System must use CDN caching for static content

3. Integration Architecture

3.1 API Integration

- **AR3.1.1:** System must implement adapter pattern for platform APIs
- **AR3.1.2:** System must handle API rate limiting and quotas
- **AR3.1.3:** System must implement retry logic for transient failures
- **AR3.1.4:** System must maintain API version compatibility

3.2 Authentication Integration

- **AR3.2.1:** System must implement OAuth flows for platform authentication
- **AR3.2.2:** System must securely store authentication tokens
- **AR3.2.3:** System must refresh tokens before expiration
- **AR3.2.4:** System must handle authentication failures gracefully

3.3 Webhook Integration

- **AR3.3.1:** System must implement webhook receivers for platform events
- **AR3.3.2:** System must validate webhook signatures
- **AR3.3.3:** System must process webhook events asynchronously
- **AR3.3.4:** System must handle webhook delivery failures

Data Requirements

1. Data Entities

1.1 User Data

- **DR1.1.1:** System must store user profile information
- **DR1.1.2:** System must store user preferences and settings
- **DR1.1.3:** System must store subscription and billing information
- **DR1.1.4:** System must store user activity logs

1.2 Content Data

- **DR1.2.1:** System must store content metadata (title, description, tags)
- **DR1.2.2:** System must store content assets (images, videos)
- **DR1.2.3:** System must store content scheduling information
- **DR1.2.4:** System must store content performance metrics

1.3 Analytics Data

- **DR1.3.1:** System must store platform-specific metrics
- **DR1.3.2:** System must store normalized cross-platform metrics
- **DR1.3.3:** System must store historical performance data
- **DR1.3.4:** System must store audience demographic data

1.4 Monetization Data

- **DR1.4.1:** System must store revenue transaction data
- **DR1.4.2:** System must store brand deal and contract information
- **DR1.4.3:** System must store affiliate link and performance data
- **DR1.4.4:** System must store financial reports and projections

1.5 Engagement Data

- **DR1.5.1:** System must store comments and messages
- **DR1.5.2:** System must store follower information and interactions
- **DR1.5.3:** System must store sentiment analysis results
- **DR1.5.4:** System must store engagement metrics and history

2. Data Relationships

2.1 User Relationships

- **DR2.1.1:** System must maintain relationships between users and content
- **DR2.1.2:** System must maintain relationships between users and platforms
- **DR2.1.3:** System must maintain relationships between users and team members
- **DR2.1.4:** System must maintain relationships between users and monetization

2.2 Content Relationships

- **DR2.2.1:** System must maintain relationships between content and platforms
- **DR2.2.2:** System must maintain relationships between content and performance
- **DR2.2.3:** System must maintain relationships between content and monetization
- **DR2.2.4:** System must maintain relationships between content and engagement

2.3 Platform Relationships

- **DR2.3.1:** System must maintain relationships between platforms and metrics
- **DR2.3.2:** System must maintain relationships between platforms and audience
- **DR2.3.3:** System must maintain relationships between platforms and monetization
- **DR2.3.4:** System must maintain relationships between platforms and content types

3. Data Retention

3.1 Short-term Retention

- **DR3.1.1:** System must retain detailed metrics for at least 90 days
- **DR3.1.2:** System must retain engagement data for at least 180 days
- **DR3.1.3:** System must retain content performance data for at least 1 year

- **DR3.1.4:** System must retain user activity logs for at least 90 days

3.2 Long-term Retention

- **DR3.2.1:** System must retain aggregated metrics indefinitely
- **DR3.2.2:** System must retain content metadata indefinitely
- **DR3.2.3:** System must retain monetization records for at least 7 years
- **DR3.2.4:** System must implement data archiving for historical data

3.3 Data Purging

- **DR3.3.1:** System must purge personal data upon account deletion
- **DR3.3.2:** System must anonymize data for analytical purposes
- **DR3.3.3:** System must implement data purging according to retention policies
- **DR3.3.4:** System must provide audit trails for data purging

Integration Requirements

1. Platform Integrations

1.1 TikTok Integration

- **IR1.1.1:** System must integrate with TikTok Creator API
- **IR1.1.2:** System must support content publishing to TikTok
- **IR1.1.3:** System must retrieve analytics data from TikTok
- **IR1.1.4:** System must monitor TikTok API changes and adapt accordingly

1.2 Instagram Integration

- **IR1.2.1:** System must integrate with Instagram Graph API
- **IR1.2.2:** System must support content publishing to Instagram Reels
- **IR1.2.3:** System must retrieve analytics data from Instagram
- **IR1.2.4:** System must monitor Instagram API changes and adapt accordingly

1.3 X Integration

- **IR1.3.1:** System must integrate with X API
- **IR1.3.2:** System must support content publishing to X
- **IR1.3.3:** System must retrieve analytics data from X
- **IR1.3.4:** System must monitor X API changes and adapt accordingly

2. Third-Party Integrations

2.1 Analytics Integrations

- **IR2.1.1:** System must integrate with Google Analytics
- **IR2.1.2:** System must support custom analytics pixel integration
- **IR2.1.3:** System must integrate with attribution tracking services
- **IR2.1.4:** System must support data export to business intelligence tools

2.2 Monetization Integrations

- **IR2.2.1:** System must integrate with major affiliate networks
- **IR2.2.2:** System must integrate with payment processors
- **IR2.2.3:** System must integrate with contract management systems
- **IR2.2.4:** System must support integration with accounting software

2.3 Content Integrations

- **IR2.3.1:** System must integrate with cloud storage services
- **IR2.3.2:** System must integrate with digital asset management systems
- **IR2.3.3:** System must support integration with content creation tools
- **IR2.3.4:** System must integrate with stock media libraries

3. API Requirements

3.1 Public API

- **IR3.1.1:** System must provide RESTful API for third-party integration
- **IR3.1.2:** API must implement proper authentication and authorization
- **IR3.1.3:** API must implement rate limiting and usage quotas
- **IR3.1.4:** API must be versioned to support backward compatibility

3.2 Webhook Support

- **IR3.2.1:** System must provide webhook notifications for key events
- **IR3.2.2:** Webhooks must include relevant event data
- **IR3.2.3:** Webhooks must implement security best practices
- **IR3.2.4:** System must provide webhook configuration and management

3.3 Single Sign-On

- **IR3.3.1:** System must support OAuth 2.0 for authentication
- **IR3.3.2:** System must support SAML for enterprise SSO
- **IR3.3.3:** System must support social login (Google, Apple)
- **IR3.3.4:** System must implement proper session management

Security Requirements

1. Authentication and Authorization

1.1 User Authentication

- **SR1.1.1:** System must implement email/password authentication
- **SR1.1.2:** System must support multi-factor authentication
- **SR1.1.3:** System must enforce strong password policies
- **SR1.1.4:** System must implement account lockout after failed attempts

1.2 Authorization

- **SR1.2.1:** System must implement role-based access control
- **SR1.2.2:** System must enforce principle of least privilege
- **SR1.2.3:** System must validate authorization for all requests
- **SR1.2.4:** System must log access control decisions

1.3 API Security

- **SR1.3.1:** API must use OAuth 2.0 for authentication
- **SR1.3.2:** API must implement token-based authorization
- **SR1.3.3:** API must validate all input parameters
- **SR1.3.4:** API must implement rate limiting to prevent abuse

2. Data Protection

2.1 Encryption

- **SR2.1.1:** System must encrypt all data in transit using TLS 1.3
- **SR2.1.2:** System must encrypt sensitive data at rest
- **SR2.1.3:** System must implement proper key management
- **SR2.1.4:** System must rotate encryption keys periodically

2.2 Data Privacy

- **SR2.2.1:** System must implement data minimization principles
- **SR2.2.2:** System must provide data export functionality
- **SR2.2.3:** System must support data deletion requests
- **SR2.2.4:** System must implement privacy by design principles

2.3 Secure Development

- **SR2.3.1:** System must implement secure coding practices
- **SR2.3.2:** System must undergo regular security testing
- **SR2.3.3:** System must implement proper error handling
- **SR2.3.4:** System must validate all user inputs

3. Compliance

3.1 Regulatory Compliance

- **SR3.1.1:** System must comply with GDPR requirements
- **SR3.1.2:** System must comply with CCPA requirements
- **SR3.1.3:** System must comply with industry-specific regulations
- **SR3.1.4:** System must maintain compliance documentation

3.2 Audit and Logging

- **SR3.2.1:** System must log all security-relevant events
- **SR3.2.2:** Logs must be stored securely and tamper-proof
- **SR3.2.3:** System must implement log retention policies
- **SR3.2.4:** System must support log analysis for security monitoring

3.3 Incident Response

- **SR3.3.1:** System must implement security incident detection
- **SR3.3.2:** System must have documented incident response procedures
- **SR3.3.3:** System must support forensic investigation
- **SR3.3.4:** System must implement breach notification procedures

Performance Requirements

1. Response Time

1.1 User Interface

- **PR1.1.1:** Dashboard must load within 2 seconds
- **PR1.1.2:** Page transitions must complete within 300ms
- **PR1.1.3:** Interactive elements must respond within 100ms
- **PR1.1.4:** Data visualizations must render within 1 second

1.2 API Performance

- **PR1.2.1:** API requests must complete within 500ms for 95% of requests
- **PR1.2.2:** API must handle at least 100 requests per second per user
- **PR1.2.3:** API must implement pagination for large data sets
- **PR1.2.4:** API must optimize response payload size

1.3 Background Processing

- **PR1.3.1:** Content scheduling must process within 5 seconds
- **PR1.3.2:** Analytics processing must complete within 1 minute
- **PR1.3.3:** Report generation must complete within 2 minutes
- **PR1.3.4:** Bulk operations must provide progress indicators

2. Scalability

2.1 User Scalability

- **PR2.1.1:** System must support at least 100,000 concurrent users
- **PR2.1.2:** System must support at least 1 million total users
- **PR2.1.3:** System must implement horizontal scaling for user growth
- **PR2.1.4:** System must maintain performance under increasing user load

2.2 Data Scalability

- **PR2.2.1:** System must handle at least 10TB of content data
- **PR2.2.2:** System must handle at least 100TB of analytics data
- **PR2.2.3:** System must implement data partitioning for scalability
- **PR2.2.4:** System must optimize database queries for large datasets

2.3 Transaction Scalability

- **PR2.3.1:** System must handle at least 1,000 content items per user
- **PR2.3.2:** System must handle at least 10,000 analytics records per day per user
- **PR2.3.3:** System must handle at least 100,000 engagement interactions per day per user
- **PR2.3.4:** System must implement efficient batch processing

3. Availability

3.1 Uptime

- **PR3.1.1:** System must maintain 99.9% uptime
- **PR3.1.2:** Planned maintenance must not exceed 4 hours per month
- **PR3.1.3:** System must implement zero-downtime deployments
- **PR3.1.4:** System must provide status page for service health

3.2 Fault Tolerance

- **PR3.2.1:** System must implement redundancy for critical components
- **PR3.2.2:** System must handle component failures gracefully
- **PR3.2.3:** System must implement circuit breakers for external dependencies
- **PR3.2.4:** System must implement fallback mechanisms for critical features

3.3 Disaster Recovery

- **PR3.3.1:** System must implement automated backups
- **PR3.3.2:** System must support point-in-time recovery
- **PR3.3.3:** System must have documented disaster recovery procedures

- **PR3.3.4:** System must test recovery procedures regularly

Compliance Requirements

1. Data Protection

1.1 GDPR Compliance

- **CR1.1.1:** System must implement data subject access requests
- **CR1.1.2:** System must implement right to be forgotten
- **CR1.1.3:** System must implement data portability
- **CR1.1.4:** System must maintain records of processing activities

1.2 CCPA Compliance

- **CR1.2.1:** System must disclose data collection practices
- **CR1.2.2:** System must provide opt-out mechanisms
- **CR1.2.3:** System must implement data deletion requests
- **CR1.2.4:** System must maintain compliance documentation

1.3 Industry Compliance

- **CR1.3.1:** System must comply with platform terms of service
- **CR1.3.2:** System must comply with advertising standards
- **CR1.3.3:** System must implement age verification where required
- **CR1.3.4:** System must comply with intellectual property laws

2. Accessibility

2.1 WCAG Compliance

- **CR2.1.1:** System must comply with WCAG 2.1 AA standards
- **CR2.1.2:** System must support screen readers
- **CR2.1.3:** System must implement keyboard navigation
- **CR2.1.4:** System must provide sufficient color contrast

2.2 Internationalization

- **CR2.2.1:** System must support multiple languages
- **CR2.2.2:** System must handle different date and time formats
- **CR2.2.3:** System must support currency localization
- **CR2.2.4:** System must handle right-to-left languages

3. Legal Compliance

3.1 Terms of Service

- **CR3.1.1:** System must have clear terms of service
- **CR3.1.2:** System must obtain user consent for terms
- **CR3.1.3:** System must maintain version history of terms
- **CR3.1.4:** System must notify users of material changes

3.2 Privacy Policy

- **CR3.2.1:** System must have comprehensive privacy policy
- **CR3.2.2:** System must obtain consent for data collection
- **CR3.2.3:** System must disclose third-party data sharing
- **CR3.2.4:** System must update policy for material changes

3.3 Intellectual Property

- **CR3.3.1:** System must respect copyright of content
- **CR3.3.2:** System must implement DMCA compliance procedures
- **CR3.3.3:** System must protect user intellectual property
- **CR3.3.4:** System must have clear content ownership policies

Assumptions and Constraints

1. Assumptions

1.1 Technical Assumptions

- **AC1.1.1:** Users have reliable internet connectivity
- **AC1.1.2:** Platform APIs will remain stable with reasonable notice for changes
- **AC1.1.3:** Users have modern web browsers
- **AC1.1.4:** Cloud infrastructure will be available and reliable

1.2 Business Assumptions

- **AC1.2.1:** Content creators will continue to use multiple platforms
- **AC1.2.2:** Platform monetization options will continue to expand
- **AC1.2.3:** Creator economy will continue to grow
- **AC1.2.4:** Users will be willing to pay for productivity improvements

1.3 User Assumptions

- **AC1.3.1:** Users have basic technical proficiency
- **AC1.3.2:** Users understand social media metrics
- **AC1.3.3:** Users have established accounts on platforms
- **AC1.3.4:** Users create content regularly

2. Constraints

2.1 Technical Constraints

- **AC2.1.1:** Platform API limitations and rate limits
- **AC2.1.2:** Platform policy restrictions on automation
- **AC2.1.3:** Browser compatibility requirements
- **AC2.1.4:** Mobile device performance limitations

2.2 Business Constraints

- **AC2.2.1:** Development budget and timeline
- **AC2.2.2:** Pricing strategy must align with market expectations
- **AC2.2.3:** Must differentiate from existing solutions
- **AC2.2.4:** Must comply with platform partner programs

2.3 Legal Constraints

- **AC2.3.1:** Data protection regulations
- **AC2.3.2:** Platform terms of service compliance
- **AC2.3.3:** Intellectual property considerations
- **AC2.3.4:** International legal requirements

Success Metrics

1. User Success Metrics

1.1 Adoption Metrics

- **SM1.1.1:** Number of registered users
- **SM1.1.2:** User growth rate
- **SM1.1.3:** Platform connections per user
- **SM1.1.4:** Feature adoption rates

1.2 Engagement Metrics

- **SM1.2.1:** Daily active users
- **SM1.2.2:** Weekly active users
- **SM1.2.3:** Session duration
- **SM1.2.4:** Feature usage frequency

1.3 Satisfaction Metrics

- **SM1.3.1:** Net Promoter Score
- **SM1.3.2:** Customer satisfaction ratings
- **SM1.3.3:** Support ticket volume and resolution time
- **SM1.3.4:** Feature request volume and themes

2. Business Success Metrics

2.1 Revenue Metrics

- **SM2.1.1:** Monthly recurring revenue
- **SM2.1.2:** Annual recurring revenue
- **SM2.1.3:** Average revenue per user
- **SM2.1.4:** Revenue growth rate

2.2 Retention Metrics

- **SM2.2.1:** Monthly churn rate
- **SM2.2.2:** Annual retention rate
- **SM2.2.3:** Subscription upgrade rate
- **SM2.2.4:** Winback rate for churned users

2.3 Efficiency Metrics

- **SM2.3.1:** Customer acquisition cost
- **SM2.3.2:** Lifetime value
- **SM2.3.3:** LTV:CAC ratio
- **SM2.3.4:** Payback period

3. Product Success Metrics

3.1 Performance Metrics

- **SM3.1.1:** System uptime
- **SM3.1.2:** Average response time
- **SM3.1.3:** Error rate
- **SM3.1.4:** API reliability

3.2 Quality Metrics

- **SM3.2.1:** Bug density
- **SM3.2.2:** Test coverage
- **SM3.2.3:** Technical debt metrics
- **SM3.2.4:** Security vulnerability count

3.3 Development Metrics

- **SM3.3.1:** Sprint velocity
- **SM3.3.2:** Release frequency
- **SM3.3.3:** Time to implement features
- **SM3.3.4:** Time to resolve issues

Future Considerations

1. Platform Expansion

1.1 Additional Platforms

- **FC1.1.1:** YouTube integration
- **FC1.1.2:** LinkedIn integration
- **FC1.1.3:** Pinterest integration
- **FC1.1.4:** Emerging platform integration

1.2 Content Types

- **FC1.2.1:** Long-form video support
- **FC1.2.2:** Live streaming management
- **FC1.2.3:** Podcast distribution
- **FC1.2.4:** Newsletter integration

1.3 Geographic Expansion

- **FC1.3.1:** Localization for international markets
- **FC1.3.2:** Region-specific platform support
- **FC1.3.3:** International compliance requirements
- **FC1.3.4:** Multi-currency support

2. Feature Expansion

2.1 Advanced Analytics

- **FC2.1.1:** Predictive analytics for content performance
- **FC2.1.2:** Competitive intelligence features
- **FC2.1.3:** Advanced audience segmentation
- **FC2.1.4:** Attribution modeling

2.2 AI Capabilities

- **FC2.2.1:** AI-generated content suggestions
- **FC2.2.2:** Automated content repurposing
- **FC2.2.3:** Intelligent scheduling optimization
- **FC2.2.4:** Automated response generation

2.3 Collaboration Features

- **FC2.3.1:** Agency/client collaboration tools
- **FC2.3.2:** Creator collaboration marketplace
- **FC2.3.3:** Brand partnership matching
- **FC2.3.4:** Content approval workflows

3. Business Model Evolution

3.1 Pricing Tiers

- **FC3.1.1:** Enterprise tier for agencies and large teams
- **FC3.1.2:** Freemium tier for emerging creators
- **FC3.1.3:** Usage-based pricing options
- **FC3.1.4:** Custom pricing for enterprise clients

3.2 Revenue Streams

- **FC3.2.1:** Marketplace commission model
- **FC3.2.2:** Premium features and add-ons
- **FC3.2.3:** Data and insights products
- **FC3.2.4:** White-label solutions

3.3 Partnership Opportunities

- **FC3.3.1:** Platform partnership programs
- **FC3.3.2:** Integration with creator education
- **FC3.3.3:** Brand partnership network
- **FC3.3.4:** Agency reseller program

Appendices

Appendix A: Glossary

Term	Definition
Creator	Individual who produces content for social media platforms
Platform	Social media service where content is published (TikTok, Instagram, X)
Engagement	User interactions with content (likes, comments, shares)
Monetization	Methods for generating revenue from content
Analytics	Data and insights about content and audience performance

Appendix B: User Research Summary

This PRD is informed by research conducted with 50+ content creators across different follower tiers. Key findings include:

1. Creators spend an average of 15 hours per week on platform management tasks
2. 78% of creators use at least 3 different tools to manage their content
3. 92% report challenges with cross-platform analytics
4. 85% struggle with monetization tracking across multiple revenue streams
5. 73% report burnout related to content creation demands

Appendix C: Competitive Analysis

Competitor	Strengths	Weaknesses
Buffer	Simple scheduling, affordable	Limited analytics, no monetization features
Hootsuite	Comprehensive social management	Not creator-focused, expensive
Later	Strong visual planning tools	Limited to Instagram focus, basic analytics
Sprout Social	Enterprise-grade analytics	Designed for brands, not creators
Creator IQ	Comprehensive creator management	Designed for brands managing creators, not for creators themselves

Appendix D: API Dependencies

Platform	API	Features Used	Limitations
TikTok	Creator API	Content publishing, analytics	Rate limits, limited metadata
Instagram	Graph API	Content publishing, analytics	Complex authentication, limited Reels API
X	API v2	Content publishing, analytics	Rate limits, limited media support

Appendix E: Regulatory Considerations

1. **GDPR:** Impacts data collection, storage, and processing for EU users
2. **CCPA:** Impacts data collection and privacy for California users
3. **Platform Policies:** Each platform has specific terms of service that restrict automation and data usage
4. **FTC Guidelines:** Disclosure requirements for sponsored content
5. **Intellectual Property:** Copyright considerations for content creation and management