Software Requirement Specifications

Version: [1.5]

Project ID	F25-220
Supervisor	Sir Basit Ali Jasani
Co Supervisor	Dr. Muhammad Rafi
Project Team	Syed Uzair Hussain - 22K4212 Saad Ahmed - 22K4345 Huzaifa Bin Khalid - 22K4223
Submission Date	7-12-2025

[Instructions]

- No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
- All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
- This’ Instruction’ section should also be removed in final document.
- MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

Version	Name of Person	Date	Description of change
1.0	Saad Ahmed	11/29/25	Document Created
1.0	Saad Ahmed	11/29/25	Added System Description and Interface Requirements
1.1	Huzaifa Bin Khalid	11/30/25	Added Functional Requirements section
1.2	Syed Uzair Hussain	12/01/25	Added Use Case Diagrams and descriptions
1.3	Syed Uzair Hussain	12/02/25	Added Non-Functional Requirements
1.4	Huzaifa Bin Khalid	12/02/25	Added System Interfaces section
1.5	Saad Ahmed	12/06/25	Final review and formatting

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

Name	Role
Sir Basit Ali Jasani	Supervisor
Dr. Muhammad Rafi	Co- Supervisor

Document Sign-Off

[Following table will contain sign-off details of the document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

Version	Sign-off Authority	Sign-off Date
1.5	Supervisor	12/6/2025

Table of Contents

1. Introduction

1.1. Purpose of Document

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed and structured outline of the functional and non-functional requirements for **CloutCheck AI**, a multimodal reputation risk analysis platform. This document serves as a comprehensive guide for all stakeholders, including business users, developers, and technical teams, to ensure a shared understanding of the system's goals, functionalities, and constraints.

This document is:

- To provide a clear communication framework among stakeholders
- To act as a reference document for developers during the system's design, implementation, and testing phases.
- To ensure that the system's design and implementation align with the business objectives
- To establish a foundation for requirement traceability throughout the project lifecycle

1.2. Intended Audience

- Fast NU
- Jury
- Supervisor
- Co-Supervisor
- Students of Fast NU
- Our Team (Designers, Developers, Testers)
- Potential Users (Brand Managers, HR Professionals, Agency Representatives, Parents/Guardians)
- Potential Users of this product

1.3. Abbreviations

- AI = Artificial Intelligence
- API = Application Programming Interface
- BERT = Bidirectional Encoder Representations from Transformers
- CLIP = Contrastive Language-Image Pre-training
- CV = Computer Vision
- ML = Machine Learning
- NLP = Natural Language Processing
- NSFW = Not Safe For Work
- RBAC = Role-Based Access Control
- ResNet = Residual Neural Network
- RoBERTa = Robustly Optimized BERT Pretraining Approach
- SaaS = Software as a Service
- SDK = Software Development Kit
- ViT = Vision Transformer
- YOLO = You Only Look Once

1.4. Document Convention

- Font Family = Arial
- Font Size = 12 for headings, 10 for the rest of the content
- Priorities = High, Medium, Low (where applicable)
- Text Formatting =
  - Bold text indicates headings and key terms
  - Italic text indicates emphasis
  - Code or technical terms are displayed in `monospace font`

2. Overall System Description

2.1. Project Background

In today's hyper-connected digital landscape, reputation has become one of the most valuable assets for individuals and organizations alike. Social media platforms have transformed influencers and public figures into powerful agents of brand visibility, marketing reach, and public perception. For brands, a single controversial post, video, or statement from their influencer partner can rapidly spiral into reputational damage, financial loss, and diminished trust.

However, the same platforms that amplify brand reach also expose organizations to significant reputational risks. Existing tools for influencer vetting and social listening primarily focus on sentiment tracking, follower metrics, and keyword monitoring, but they lack the capability to evaluate content across diverse modalities (text, images, videos, audio) and consolidate findings into a unified risk score.

Problem Statement:

- Manual vetting of influencers is time-consuming, inconsistent, and prone to human bias
- Text-only sentiment analysis fails to capture harmful visual, video, or audio content
- No standardized, objective method exists for quantifying influencer reputation risk
- Brands cannot track reputation trends over time or across platforms effectively
- Multimodal content (the dominant form of social media) requires specialized AI models

This gap underscores the pressing need for an intelligent, automated, and multimodal reputation risk assessment platform that can help organizations make informed, data-driven partnership decisions.

2.2. Project Scope

CloutCheck AI aims to streamline organizational influencer assessment using AI and multimodal analysis technologies, ensuring transparency, efficiency, and fairness. The primary goal is to provide an integrated platform for Brand Managers, HR Professionals, Agencies, and Parents to manage influencer evaluation, content analysis, and reputation risk scoring effectively.

Included Functionalities:

1. User Management & Authentication

- Secure registration and login with role-based access (Admin, Brand Manager, HR Professional, Agency User, Parent/Guardian)
- Profile management and user preferences
- Password reset and account recovery

2. Profile Integration & Content Acquisition

- Manual input of influencer social media handles (Instagram, TikTok, Twitter/X, YouTube)
- Automated content scraping using official APIs and controlled web scraping
- Support for multiple platform profiles per influencer
- Historical content collection with configurable date ranges

3. Multimodal Content Analysis

- **Text Analysis**: NLP-based detection of hate speech, toxicity, extremism, offensive language, and sentiment analysis using BERT/RoBERTa models
- **Image Analysis**: Computer vision models (YOLOv8/ResNet) to identify NSFW content, violence, weapons, drugs, alcohol, and inappropriate imagery
- **Video Analysis**: Frame-by-frame extraction and analysis with temporal tracking for harmful elements
- **Audio/Speech Analysis**: Speech-to-text transcription using Whisper, followed by toxic language detection and sentiment analysis

4. Multimodal Fusion & Risk Scoring

- Integration of insights from all modalities using fusion strategies (early fusion, late fusion, attention-guided fusion)
- Risk classification across categories: hate speech, sexual/NSFW content, violence, drugs/alcohol, extremism, political controversy, misinformation
- Generation of overall Reputation Risk Score (0-100 scale) with confidence measures
- Category-specific severity scoring

5. Decision Support & Reporting

- Interactive web-based dashboard with real-time visualization of risk metrics
- Trend analysis and timeline visualization of risk patterns
- Flagged content gallery with evidence (text snippets, image thumbnails, video timestamps, audio excerpts)
- Professional, branded PDF report generation with actionable insights
- Comparison tools for evaluating multiple influencers
- Alert system for high-risk content detection

6. Real-Time Monitoring (Optional/Future Enhancement)

- Continuous monitoring of influencer profiles for new content
- Automated alerts for newly detected risks

- Scheduled recurring scans

2.3. Not In Scope

The following functionalities are explicitly excluded from the current version of CloutCheck AI:

- **Private/Protected Content Access**: The system will not access private accounts, password-protected content, or content behind paywalls
- **Live Streaming Analysis**: Real-time analysis of ongoing live streams is not supported
- **Direct Platform Moderation**: The system does not remove or moderate content on social media platforms
- **Legal Enforcement**: CloutCheck AI provides risk assessment only and does not take legal action or enforce platform policies
- **Fake Profile Detection**: Identifying bot accounts or fake followers is not within scope
- **Content Creation/Editing**: The system does not generate, edit, or post content
- **Financial Transaction Processing**: No payment processing or e-commerce functionality
- **Mobile Application**: Current scope includes web application only (mobile app is future enhancement)
- **Influencer Performance Analytics**: Engagement rates, follower growth, and ROI metrics are not included
- **Manual Content Upload**: Users cannot upload content directly; all content is scraped from platforms

2.4. Project Objectives

The aim of this project is to design and develop an AI-powered multimodal reputation risk analysis platform that automatically collects and analyzes digital content from social media and online sources, evaluates reputational risks across multiple categories, and generates an objective, transparent reputation score to support informed decision-making by brands, organizations, and individuals.

Specific Objectives:

1. **Automated Content Acquisition and Integration**
   - Develop automated methods for extracting user-generated content from social media platforms using official APIs and custom web scrapers
   - Ensure secure handling of data through role-based authentication and privacy-preserving mechanisms
   - Support integration with at least 3-4 major platforms (Instagram, TikTok, Twitter/X, YouTube)
2. **Multimodal Content Analysis**
   - Implement natural language processing techniques to detect sentiment, hate speech, extremism, and offensive language in textual data

- Apply computer vision models to identify high-risk elements such as nudity, drugs, violence, and weapons in images and video frames
- Incorporate speech-to-text transcription and offensive language detection for audio and video sources
- Achieve benchmark accuracy levels for each modality-specific classifier

3. **Reputation Risk Scoring**

   - Design a transparent, standardized scoring framework that assigns severity levels (0-100) across multiple risk categories
   - Integrate modality-specific outputs into a unified reputation index with confidence measures
   - Provide explainable AI features showing evidence and reasoning behind risk scores

4. **Decision Support and Visualization**

   - Develop an interactive dashboard for real-time visualization of reputation metrics, trend analysis, and alerts
   - Implement branded PDF report generation with actionable insights for decision-makers
   - Provide filtering, sorting, and comparison capabilities for analyzing multiple profiles

5. **System Deployment and Evaluation**

   - Deploy the platform on a scalable cloud-based infrastructure with support for multi-user access
   - Evaluate system performance using accuracy benchmarks, usability testing, and comparative analysis with existing solutions
   - Ensure 99.9% uptime and response times under 3 seconds for standard queries

2.5. Stakeholders

- Brand Managers
- Influencer Agencies
- Legal/Compliance Teams
- Project Team
- Supervisor & Co-Supervisor
- End Users (General)
- HODs
- HR
- Organization (Fast NU)

2.6. Operating Environment

Hardware Environment:

- **Client-Side**: Desktop computers, laptops, tablets with modern web browsers
  - Minimum: 4GB RAM, dual-core processor
  - Recommended: 8GB+ RAM, quad-core processor
  - Display: 1366x768 minimum resolution (1920x1080 recommended)
- **Server-Side**: Cloud-based infrastructure (AWS/GCP/Azure)
  - Compute: Scalable instances with GPU support for AI model inference

- Storage: Object storage (S3/Cloud Storage) for media files, managed database services for metadata
- Network: Load balancers, CDN for static assets

Software Environment:

- **Client-Side**:
  - Modern web browsers: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
  - JavaScript enabled
  - Internet connection required
- **Server-Side**:
  - Operating System: Linux (Ubuntu 20.04+ or CentOS 8+)
  - Python 3.9+ runtime environment
  - Node.js 16+ for frontend build tools
  - MongoDB 5.0+ for database
  - Redis 6.0+ for caching
  - Docker and Kubernetes for containerization and orchestration
- **Development Environment**:
  - Git for version control
  - VS Code/PyCharm for development
  - Postman for API testing

Network Environment:

- Stable internet connectivity required for both users and servers
- HTTPS/TLS 1.3 for all communications
- API rate limiting and throttling mechanisms
- CDN for global content delivery

2.7. System Constraints

Technical Constraints:

1. **AI Model Constraints**:
   - Pre-trained models must be used and fine-tuned due to limited computational resources for training from scratch
   - Model inference latency must be optimized to ensure response times under 30 seconds for standard scans
   - GPU availability may limit concurrent processing capacity
2. **API and Platform Constraints**:
   - Social media platforms impose rate limits on API calls (e.g., Instagram Graph API, Twitter API v2)
   - Some platforms have restricted or deprecated public APIs (e.g., Instagram)
   - Web scraping must respect robots.txt and terms of service to avoid legal issues

  - Only publicly available content can be accessed (no private accounts or protected content)

3. **Data Storage Constraints**:
   - Academic/free-tier cloud resources impose storage limits
   - Media files (images, videos) must be stored efficiently with compression where possible
   - Database query performance must be optimized for large-scale data
4. **Processing Constraints**:
   - Video processing is computationally intensive; frame sampling strategies must balance accuracy and performance
   - Audio transcription for long videos may take significant time
   - Real-time monitoring features may require significant infrastructure investment
5. **Language Constraints**:
   - Primary focus on English language content
   - Multilingual support (especially code-mixed languages like Urdu-English) requires specialized models
   - Non-English content may have reduced accuracy without language-specific model training

Operational Constraints:

1. **Budget Constraints**:
   - Limited to academic project budget
   - Reliance on free-tier services, student credits, and open-source tools
   - Commercial API usage (e.g., Apify for scraping) may be limited
2. **Time Constraints**:
   - Development must be completed within two semesters (9 months)
   - Phased delivery with MVP focus
3. **Legal and Ethical Constraints**:
   - Must comply with data protection regulations (GDPR-like principles)
   - Cannot access or store personally identifiable information beyond public usernames
   - Must respect platform terms of service
   - Cannot be used for surveillance or stalking purposes
   - AI models must be evaluated for bias and fairness
4. **User Interface Constraints**:
   - Web-only interface (no mobile app in current scope)
   - Must support multiple screen sizes (responsive design)
   - Accessibility standards (WCAG 2.1 Level AA) should be considered

Cultural and Context Constraints:

1. Cultural sensitivity in risk assessment (what is considered offensive varies by region)
2. Context-dependent interpretation of content (sarcasm, humor, cultural references)

3. Rapidly evolving social media trends and slang may challenge static models

2.8. Assumptions

1. **User Behavior**:
   - Users will provide accurate influencer handles and profile information
   - Users have basic technical literacy to navigate a web application
   - Users understand that the system provides risk assessment, not absolute judgment
   - Users will use the system ethically and legally
2. **Data Availability**:
   - Influencers maintain public social media profiles
   - Social media platforms will continue to allow access to public content via APIs or web scraping
   - Sufficient historical content exists for meaningful analysis (at least 20-30 posts)
3. **Technical Infrastructure**:
   - Cloud computing resources (AWS/GCP/Azure free tier or student credits) will be available
   - Internet connectivity will be stable during development and deployment
   - Open-source AI models and libraries will remain available and maintained
4. **Model Performance**:
   - Pre-trained AI models (BERT, RoBERTa, YOLOv8, Whisper) will achieve acceptable baseline accuracy (70%+ on benchmark datasets)
   - Fine-tuning on domain-specific data will improve performance
   - Multimodal fusion will yield better results than single-modality analysis
5. **Stakeholder Engagement**:
   - Supervisor and co-supervisor will provide regular guidance
   - Potential end users (brand managers, HR professionals) will be available for feedback during development.

2.9. Dependencies

1. **External APIs and Services**:
   - **Social Media APIs**: Instagram Graph API, Twitter API v2, TikTok API, YouTube Data API
   - **Scraping Services**: Apify, Bright Data, or custom scraping tools (subject to platform ToS)
   - **CrossRef API**: For research paper verification (if integrating academic credentials in future)
   - Dependency Risk: API deprecation, rate limit changes, or pricing changes could impact functionality
2. **AI/ML Frameworks and Libraries**:
   - **PyTorch** or **TensorFlow**: Core deep learning frameworks
   - **Hugging Face Transformers**: Pre-trained NLP models (BERT, RoBERTa)
   - **OpenCV**: Video and image processing
   - **Whisper**: Speech-to-text transcription
   - **YOLO/Ultralytics**: Object detection models
   - Dependency Risk: Breaking changes in library versions, model availability
3. **Cloud Infrastructure**:
   - **AWS/GCP/Azure**: Compute, storage, and database services
   - **MongoDB Atlas**: Managed database service
   - **Redis Cloud**: Caching service
   - **CDN**: Content delivery for static assets
   - Dependency Risk: Service outages, pricing changes, free-tier limitations
4. **Development Tools**:
   - **Git/GitHub**: Version control and collaboration

- **Docker**: Containerization for deployment
- **FastAPI/Flask**: Backend web framework
- **React/Next.js**: Frontend framework
- **Material-UI/Tailwind CSS**: UI component libraries

5. **Data Sources**:
   - **Training Datasets**: HateXplain (text toxicity), MuTox (audio toxicity), NSFW datasets, violence detection datasets
   - **Benchmark Datasets**: For model evaluation and validation
   - Dependency Risk: Dataset availability, licensing restrictions, data quality issues
6. **Third-Party Documentation**:
   - Official API documentation from social media platforms
   - Model documentation from Hugging Face, OpenAI, and other providers
   - Dependency Risk: Documentation changes, lack of support for edge cases

Mitigation Strategies:

- Maintain fallback options for critical dependencies (e.g., multiple scraping methods)
- Version control all dependencies with lock files
- Regular monitoring of API status and deprecation notices
- Community engagement with open-source projects for support
- Budget allocation for critical paid services if free tiers are insufficient

3. External Interface Requirements

3.1. Hardware Interfaces

Client-Side Hardware:

- **Desktop/Laptop Computers**:
    - Minimum: 4GB RAM, dual-core processor (Intel i3 or equivalent)
    - Recommended: 8GB+ RAM, quad-core processor (Intel i5/AMD Ryzen 5 or better)
    - Display: 1366x768 minimum resolution (1920x1080 recommended)
    - Input: Standard keyboard and mouse/trackpad
    - Network: Wired or wireless internet connection (minimum 5 Mbps)
- **Mobile Devices** (Future Enhancement):
    - Not supported in current version but responsive design will allow basic access

Server-Side Hardware:

- **Cloud Infrastructure** (AWS EC2, GCP Compute Engine, or Azure VMs):
    - Compute Instances:
        - General Purpose: 2-4 vCPUs, 8-16GB RAM for backend API services
        - GPU Instances: NVIDIA T4/V100 GPUs for AI model inference
    - Storage:
        - Object Storage (S3/Cloud Storage): For media files, model weights, and backups
        - Block Storage: For database volumes (SSD-backed)
    - Network:
        - Load Balancers: For traffic distribution
        - NAT Gateway: For secure outbound connections

No Direct Hardware Integration:

- The system does not interface with specialized hardware devices (e.g., cameras, sensors, biometric readers)
- All interactions are through standard web browsers and cloud infrastructure

3.2. Software Interfaces

1. Social Media Platform APIs:

Instagram Graph API / Instagram Basic Display API:

- **Interface Type**: RESTful API over HTTPS
- **Data Format**: JSON
- **Authentication**: OAuth 2.0 with access tokens
- **Operations**:
    - Retrieve user profile information (username, bio, profile picture)
    - Fetch media (photos, videos, captions)

- Access public posts and stories (if available)

- **Rate Limits**: 200 calls per hour per user (Graph API)
- **Data Exchange**:
    - Request: HTTP GET with access token
    - Response: JSON payload with media URLs, captions, metadata
- **Version**: Instagram Graph API v21.0 (or latest)

Twitter API v2 (X Platform):

- **Interface Type**: RESTful API over HTTPS
- **Data Format**: JSON
- **Authentication**: OAuth 2.0 Bearer Token
- **Operations**:
    - Retrieve tweets from user timeline
    - Fetch user profile data
    - Access tweet metadata (likes, retweets, replies)
- **Rate Limits**: Varies by endpoint (e.g., 900 requests per 15 minutes for user timeline)
- **Data Exchange**:
    - Request: HTTP GET with Bearer token in Authorization header
    - Response: JSON with tweet objects, user objects, media entities
- **Version**: Twitter API v2

2. Web Scraping Services:

Apify Platform:

- **Interface Type**: RESTful API over HTTPS
- **Data Format**: JSON
- **Authentication**: API Token
- **Operations**:
    - Run scrapers for Instagram, TikTok, Twitter, YouTube
    - Retrieve scraped data (posts, images, videos, captions)
- **Data Exchange**:
    - Request: HTTP POST to start scraping task with target profile URL
    - Response: JSON with scraped content or task ID for polling
- **Documentation**: https://apify.com/apify/instagram-scraper, https://apify.com/apify/tiktok-scraper

Custom Scrapers (Python-based):

- **Libraries**: BeautifulSoup, Selenium, Scrapy
- **Operations**: Parse HTML/JavaScript from public profile pages
- **Constraints**: Must respect robots.txt and platform ToS

3. AI/ML Model Interfaces:

Hugging Face Transformers Library:

- **Interface Type**: Python API
- **Models Used**:
    - **BERT/RoBERTa**: Text classification for hate speech, toxicity, sentiment
    - **CLIP**: Image-text alignment and zero-shot classification
    - **ViT (Vision Transformer)**: Image classification
- **Operations**:
    - Load pre-trained models
    - Tokenize input text
    - Perform inference and retrieve predictions
- **Data Exchange**:
    - Input: Raw text or preprocessed tensors
    - Output: Class probabilities, embeddings, logits
- **Version**: Transformers 4.x

OpenAI Whisper:

- **Interface Type**: Python API
- **Operations**: Speech-to-text transcription from audio/video files
- **Data Exchange**:
    - Input: Audio file path or byte stream
    - Output: Transcribed text with timestamps
- **Version**: Whisper v3 (or latest)

Ultralytics YOLOv8:

- **Interface Type**: Python API
- **Operations**: Object detection in images and video frames
- **Data Exchange**:
    - Input: Image file or video frames (NumPy arrays)
    - Output: Bounding boxes, class labels, confidence scores
- **Version**: YOLOv8 (latest)

PyTorch / TensorFlow:

- **Interface Type**: Python API
- **Operations**:
    - Load custom-trained models
    - Perform inference
    - Fine-tune models on custom datasets
- **Version**: PyTorch 2.x or TensorFlow 2.x

4. Database Management System:

MongoDB:

- **Interface Type**: MongoDB Wire Protocol over TCP
- **Client Library**: PyMongo (Python), Mongoose (Node.js)
- **Operations**:
    - CRUD operations (Create, Read, Update, Delete) for user accounts, scan metadata, risk scores
    - Aggregation pipelines for analytics
- **Data Exchange**:
    - Input: BSON documents (JSON-like)
    - Output: Query results as BSON/JSON
- **Version**: MongoDB 5.0+
- **Hosting**: MongoDB Atlas (managed cloud service) or self-hosted

5. Caching System:

Redis:

- **Interface Type**: Redis Protocol over TCP
- **Client Library**: redis-py (Python)
- **Operations**:
    - Cache API responses
    - Store session data
    - Implement request throttling/rate limiting
- **Data Exchange**:
    - Input: Key-value pairs (strings, hashes, sets)
    - Output: Cached data retrieval
- **Version**: Redis 6.0+
- **Hosting**: Redis Cloud or self-hosted

6. Cloud Storage:

AWS S3 / Google Cloud Storage / Azure Blob Storage:

- **Interface Type**: RESTful API over HTTPS
- **Client Library**: boto3 (AWS SDK for Python), google-cloud-storage, azure-storage-blob
- **Operations**:
    - Upload media files (images, videos, audio)
    - Download files for processing
    - Generate signed URLs for temporary access
- **Data Exchange**:
    - Input: Binary file data
    - Output: Object URLs, metadata
- **Version**: Latest SDKs

7. Backend Framework:

FastAPI / Flask:

- **Interface Type**: HTTP/HTTPS RESTful API
- **Operations**:
  - Expose API endpoints for frontend
  - Handle authentication and authorization
  - Orchestrate scraping, AI inference, and database operations
- **Data Exchange**:
  - Input: HTTP requests (GET, POST, PUT, DELETE) with JSON payloads
  - Output: HTTP responses with JSON data
- **Version**: FastAPI 0.100+ or Flask 2.x

8. Frontend Framework:

React / Next.js:

- **Interface Type**: JavaScript framework running in browser
- **Operations**:
  - Render UI components
  - Make API calls to backend
  - Display visualizations (charts, graphs)
- **Communication**: HTTP/HTTPS requests to backend API endpoints (JSON payloads)
- **Version**: React 18+ / Next.js 14+

9. Visualization Libraries:

Plotly / Chart.js / D3.js:

- **Interface Type**: JavaScript libraries
- **Operations**:
  - Render interactive charts and graphs
  - Display risk score distributions, timelines, heatmaps
- **Data Exchange**:
  - Input: JavaScript objects with data points
  - Output: SVG/Canvas-based visualizations in browser

10. PDF Generation:

ReportLab / WeasyPrint:

- **Interface Type**: Python library
- **Operations**: Generate branded PDF reports from HTML/templates
- **Data Exchange**:
  - Input: HTML content or Python objects
  - Output: PDF file (binary)

11. Email Service (Optional):

SendGrid / AWS SES / SMTP:

- **Interface Type**: RESTful API or SMTP protocol
- **Operations**: Send email notifications (account verification, scan completion alerts)
- **Data Exchange**:
  - Input: Email recipient, subject, body (HTML/plain text)
  - Output: Delivery status

- SMTP Servers: For email notifications.

3.3. Communications Interfaces

1. Network Protocols:

- **HTTP/HTTPS**: Primary protocol for all web communications
- **TLS 1.3**: Encryption for all data in transit
- **WebSocket** (Optional for future real-time features): For real-time notifications and live dashboard updates
- **TCP/IP**: Underlying network protocol for database and cache connections

2. API Communication Standards:

- **RESTful Architecture**: All backend APIs follow REST principles
  - HTTP methods: GET (retrieve), POST (create), PUT (update), DELETE (remove)
  - Stateless communication
  - Resource-based URLs (e.g., /api/scans/{scan_id})
- **JSON Data Format**: All API requests and responses use JSON
- **Authentication**: JWT (JSON Web Tokens) for stateless authentication
  - Token included in Authorization: Bearer <token> header
  - Token expiration: 24 hours
- **Rate Limiting**:
  - Per-user limits (e.g., 100 requests per minute)
  - Implemented using Redis-based counters

3. Data Exchange Formats:

- **JSON**: Primary format for structured data (API payloads, configuration files)
- **BSON**: MongoDB's binary JSON format for database storage
- **CSV/Excel**: Optional export format for scan results
- **PDF**: Report generation output format
- **Binary**: Media files (JPEG, PNG, MP4, MP3) transferred as raw bytes

4. Security Measures:

- **HTTPS Only**: All communications must use HTTPS (no plaintext HTTP)
- **API Authentication**: JWT tokens for user authentication
- **CORS (Cross-Origin Resource Sharing)**: Configured to allow frontend domain only
- **Input Validation**: All API inputs validated and sanitized to prevent injection attacks
- **Encryption at Rest**: Sensitive data encrypted in database (user credentials, personal info)
- **Audit Logging**: All API requests logged with user ID, timestamp, and operati

5. Browser Compatibility:

- **Supported Browsers**:
  - Google Chrome 90+
  - Mozilla Firefox 88+
  - Safari 14+
  - Microsoft Edge 90+
- **JavaScript**: ECMAScript 2020+ features
- **HTML5**: Modern semantic markup
- **CSS3**: Responsive design with Flexbox/Grid

6. Performance Optimization:

- **CDN (Content Delivery Network)**: Static assets (images, CSS, JS) served via CDN for faster load times
- **Compression**: Gzip/Brotli compression for text-based responses
- **Caching**:
  - Browser caching for static assets (Cache-Control headers)
  - Redis caching for frequently accessed data
- **Lazy Loading**: Images and media files loaded on-demand in UI
- **Pagination**: Large datasets returned in pages (e.g., 20 results per page)

7. Error Handling and Status Codes:

- **HTTP Status Codes**:
  - 200 OK: Successful request
  - 201 Created: Resource created successfully
  - 400 Bad Request: Invalid input data
  - 401 Unauthorized: Missing or invalid authentication token
  - 403 Forbidden: Insufficient permissions
  - 404 Not Found: Resource does not exist
  - 429 Too Many Requests: Rate limit exceeded
  - 500 Internal Server Error: Server-side error
  - 503 Service Unavailable: System maintenance or overload
- **Error Response Format**:

Json:

```
{
  "error": {
    "code": "INVALID_INPUT",
    "message": "The provided Instagram handle is invalid.",
    "details": { "field": "instagram_handle" }
  }
}
```

8. Monitoring and Logging:

- **Application Logs**: Structured logging (JSON format) with levels (DEBUG, INFO, WARN, ERROR)
- **Access Logs**: All API requests logged with timestamp, user ID, endpoint, status code
- **Error Tracking**: Integration with error monitoring services (e.g., Sentry) for production
- **Metrics**: System performance metrics (response times, error rates, throughput) collected for monitoring

4. Functional Requirements

4.1. Functional Hierarchy

1. User Management (Admin & Self-Service)

- 1.1 User Registration
- 1.2 User Login/Logout
- 1.3 Password Reset
- 1.4 Profile Management (Edit email, name, organization)
- 1.5 Admin: Create/Edit/Delete User Accounts
- 1.6 Admin: View All Users
- 1.7 Admin: Assign Roles (Admin, Brand Manager, HR Professional, Agency User)

2. Profile & Influencer Management

- 2.1 Add Influencer Profile (manual handle input)
- 2.2 Edit Influencer Handles
- 2.3 Delete Influencer Profile
- 2.4 View List of All Profiles
- 2.5 Search Profiles by Name/Handle

3. Content Acquisition & Scraping

- 3.1 Integrate with Social Media APIs (Instagram, Twitter/X, TikTok, YouTube)
- 3.2 Web Scraping for Platforms with Limited API Access
- 3.3 Configure Date Range for Historical Content Collection
- 3.4 Automatic Content Download (text, images, videos, audio)
- 3.5 Validate and Store Content Metadata

4. Multimodal Content Analysis

- **4.1 Text Analysis**
    - 4.1.1 Sentiment Analysis (positive, negative, neutral)
    - 4.1.2 Hate Speech Detection
    - 4.1.3 Toxicity/Offensive Language Detection

    - 4.1.4 Extremism/Political Content Detection
    - 4.1.5 Misinformation Detection (keyword-based, fact-checking)
- **4.2 Image Analysis**
    - 4.2.1 NSFW Content Detection (nudity, sexual imagery)
    - 4.2.2 Violence Detection (blood, weapons, fighting)
    - 4.2.3 Drug/Alcohol Detection (substances, paraphernalia)
    - 4.2.4 Weapon Detection (guns, knives)
    - 4.2.5 Object and Scene Recognition
- **4.3 Video Analysis**
    - 4.3.1 Frame Extraction and Sampling
    - 4.3.2 Frame-level Analysis (using image models)
    - 4.3.3 Temporal Risk Tracking (identify high-risk segments)
    - 4.3.4 Scene Context Analysis
- **4.4 Audio/Speech Analysis**
    - 4.4.1 Speech-to-Text Transcription (Whisper)
    - 4.4.2 Toxicity Detection in Speech
    - 4.4.3 Hate Speech Detection in Audio
    - 4.4.4 Extremist Language Detection
    - 4.4.5 Multilingual Support (English, Urdu, Hindi, code-mixed)

5. Multimodal Fusion & Risk Scoring

- 5.1 Integrate Results from All Modalities
- 5.2 Apply Fusion Strategy (Early/Late/Attention-Guided)
- 5.3 Calculate Category-Specific Risk Scores
    - 5.3.1 Hate Speech Score
    - 5.3.2 NSFW Score
    - 5.3.3 Violence Score
    - 5.3.4 Drugs/Alcohol Score

    - 5.3.5 Extremism Score
    - 5.3.6 Political Controversy Score
    - 5.3.7 Misinformation Score
- 5.4 Calculate Overall Reputation Risk Score (0-100 scale)
- 5.5 Assign Confidence Level to Each Score
- 5.6 Store Risk Scores in Database

6. Dashboard & Visualization

- 6.1 Display Scan Summary (overall score, status, date)
- 6.2 Visualize Risk Categories (pie chart, bar chart)
- 6.3 Display Flagged Content Gallery
    - 6.3.1 Show Text Snippets with Highlights
    - 6.3.2 Show Image Thumbnails with Bounding Boxes
    - 6.3.3 Show Video Clips with Timestamps
    - 6.3.4 Show Audio Transcripts with Highlighted Phrases
- 6.4 Timeline Visualization (risk trends over time)
- 6.5 Platform Distribution Chart (content source breakdown)
- 6.6 Filter Content by Risk Category
- 6.7 Sort Content by Severity/Date/Platform
- 6.8 Compare Multiple Influencer Profiles

7. Reporting & Export

- 7.1 Generate Professional PDF Report
    - 7.1.1 Include Executive Summary
    - 7.1.2 Include Detailed Risk Breakdown
    - 7.1.3 Include Evidence (text snippets, image thumbnails)
    - 7.1.4 Include Recommendations
- 7.2 Customize Report Branding (logo, colors)
- 7.3 Export Data to CSV/Excel

- 7.4 Download Report as PDF
- 7.5 Share Report via Email (optional)

8. Scan Management

- 8.1 Initiate New Scan
- 8.2 View Scan Progress (queued, processing, completed, failed)
- 8.3 Cancel Ongoing Scan
- 8.4 View Scan History
- 8.5 Re-run Previous Scan (refresh data)
- 8.6 Schedule Recurring Scans (optional/future)
- 8.7 Set Scan Parameters (date range, platforms, content types)

9. Alerts & Notifications

- 9.1 Real-Time Alerts for High-Risk Content Detection
- 9.2 Email Notifications for Scan Completion
- 9.3 Dashboard Notifications for Failed Scans
- 9.4 Configurable Alert Thresholds

10. System Administration

- 10.1 Monitor System Health (server status, database status)
- 10.2 View Audit Logs (user actions, API calls)
- 10.3 Manage System Settings (rate limits, feature flags)
- 10.4 View Usage Statistics (scans per day, users, API calls)
- 10.5 Backup and Restore Data

4.2. Use Case Diagram

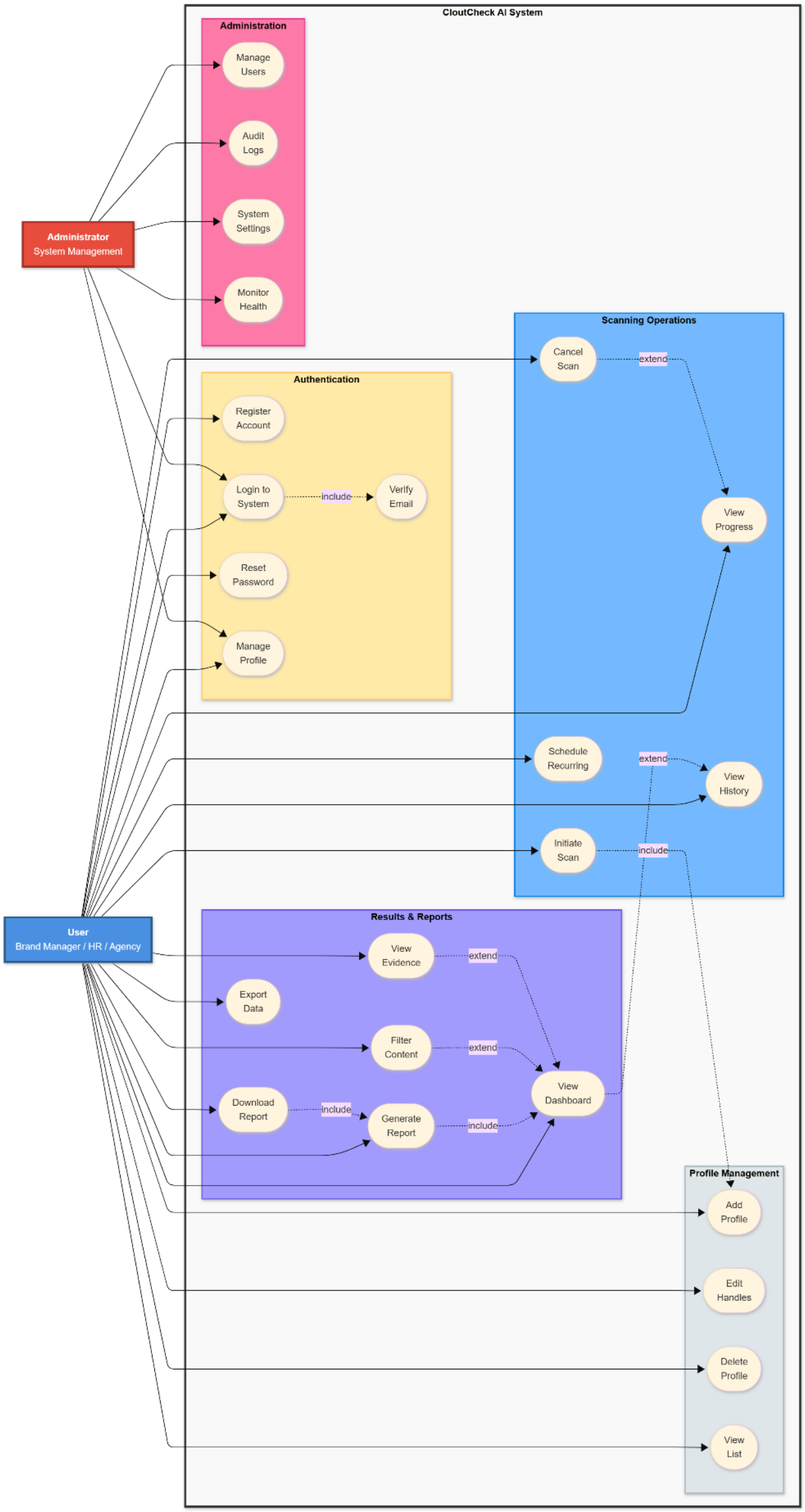

Figure 1: Use Case Diagram

Actors:

- **User** (Brand Manager / HR Professional / Agency User / Parent): Primary users who manage profiles, initiate scans, and view results
- **Administrator**: System administrator who manages users, monitors system health, and configures settings

System Modules: The system is organized into five functional modules as shown in the use case diagram:

1. **Administration** (Pink) - System management and monitoring
2. **Authentication** (Yellow) - User account and access management
3. **Scanning Operations** (Blue) - Content scanning workflows
4. **Results & Reports** (Purple) - Analysis results and reporting
5. **Profile Management** (Gray) - Influencer profile management

4.3. Module Use Cases

<table>
<tr><th colspan="3">UC-ADMIN-01: Manage Users</th></tr>
<tr><td colspan="2"><b>Use case Id:</b></td><td>UC-ADMIN-01</td></tr>
<tr><td colspan="2"><b>Actors:</b></td><td>Administrator</td></tr>
<tr><td colspan="2"><b>Feature:</b></td><td>User & Permission Management</td></tr>
<tr><td colspan="2"><b>Pre-condition:</b></td><td>Admin must be logged in.</td></tr>
<tr><td colspan="3"><b>Scenarios</b></td></tr>
<tr><td><b>Step#</b></td><td><b>Action</b></td><td><b>Software Reaction</b></td></tr>
<tr><td>1.</td><td>Admin navigates to “Manage Users”</td><td>System displays user list with actions (Add/Edit/Delete).</td></tr>
<tr><td>2.</td><td>Admin clicks “Add User”</td><td>Displays form for name, email, role, password.</td></tr>
<tr><td>3.</td><td>Admin submits form</td><td>Validates data → Creates user → Shows success message.</td></tr>
<tr><td>4.</td><td>Admin selects a user to edit</td><td>Displays user details for modification.</td></tr>
<tr><td>5.</td><td>Admin submits updated info</td><td>Validates & updates record → Shows confirmation.</td></tr>
<tr><td>6.</td><td>Admin selects a user to delete</td><td>Prompts for confirmation.</td></tr>
<tr><td>7.</td><td>Admin confirms deletion</td><td>Deletes user & updates list.</td></tr>
<tr><td>8.</td><td>Admin views Audit Logs/System Health</td><td>Shows logs and system-status metrics.</td></tr>
<tr><td colspan="3"><b>Alternate Scenarios:</b></td></tr>
<tr><td colspan="3"><b>2a: Invalid user input → System highlights erroneous fields.</b>
<b>6a: User cannot be deleted due to dependencies → System warns and stops the action.</b>
<b>8a: Health API unreachable → System shows <i>“Health metrics unavailable”</i>.</b></td></tr>
<tr><td colspan="3"><b>Post Conditions</b></td></tr>
<tr><td><b>Step#</b></td><td colspan="2"><b>Description</b></td></tr>
<tr><td>—</td><td colspan="2">User accounts are added, updated, or deleted.</td></tr>
<tr><td>—</td><td colspan="2">Audit logs reflect all actions.</td></tr>
<tr><td>—</td><td colspan="2">System health insights are shown.</td></tr>
<tr><td colspan="2"><b>Use Case Cross referenced</b></td><td><b>UC-AUTH-01: User Login</b>
<b>UC-ADMIN-02: View Audit Logs</b></td></tr>
</table>

UC-AUTH-01: Register Account		
Use case Id:	UC-AUTH-01	
Actors:	Brand Manager / HR / Agency User / Admin	
Feature:	Account Registration	
Pre-condition:	None	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User opens Registration Page	Shows registration form (Email, Password, Role).
2.	User fills out details	System validates inputs.
3.	User submits the form	Creates pending account & sends verification email.
4.	User clicks email verification link	System verifies account & activates it.
Alternate Scenarios:		
2a: Email already exists → Show *“Account already registered.”* 3a: Email server issue → Show *“Verification email failed to send.”* 4a: Verification link expired → Show option to resend.		
Post Conditions		
Step#	**Description**	
—	User account is created and verified.	
—	User can log in based on assigned role.	
Use Case Cross referenced	**UC-AUTH-02: Login to System**	

UC-AUTH-02: Login to System		
Use case Id:	UC-AUTH-02	
Actors:	All Users	
Feature:	Secure Login	
Pre-condition:	User must have a verified account	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User enters email & password	System validates credentials.
2.	User clicks Login	JWT token is issued → Redirect to dashboard.
Alternate Scenarios:		
1a: Invalid credentials → Show *“Incorrect email/password.”* 2a: Account suspended → Show *“Contact admin.”*		
Post Conditions		
Step#	**Description**	
—	User is authenticated and gains access based on role.	
Use Case Cross referenced	**UC-AUTH-03: Reset Password**	

UC-AUTH-03: Reset Password

Use case Id:	UC-AUTH-03
Actors:	All Users
Feature:	Password Reset
Pre-condition:	Valid registered email

Scenarios

Step#	Action	Software Reaction
1.	User clicks “Forgot Password”	Shows reset form.
2.	User enters email	System sends password reset link.
3.	User opens link	System displays password reset page.
4.	User sets new password	Password updated & confirmation shown.

Alternate Scenarios:

2a: Email not found → Show *“Email not registered.”*
4a: Weak password → Show strength error.

Post Conditions

Step#	Description
—	User password is updated successfully.

Use Case Cross referenced	—

UC-PROFILE-01: Add Influencer Profile

Use case Id:	UC-PROFILE-01
Actors:	Brand Manager / HR / Agency User
Feature:	Influencer Profile Onboarding
Pre-condition:	User must be logged in

Scenarios

Step#	Action	Software Reaction
1.	User selects “Add Profile”	Displays handle input form.
2.	User enters social media handles	Validates platform availability & format.
3.	User submits	Profile saved → Shows confirmation.

Alternate Scenarios:

2a: Private or inaccessible profiles → Error *“Profile must be public.”*
3a: Duplicate profile → *“Profile already exists.”*

Post Conditions

Step#	Description
—	Profile is stored and ready for scanning.

Use Case Cross referenced	—

UC-PROFILE-02: Edit Influencer Profile

Use case Id:	UC-PROFILE-02
Actors:	Brand Manager / HR / Agency User
Feature:	Influencer Profile Onboarding
Pre-condition:	Influencer profile must exist

Scenarios

Step#	**Action**	**Software Reaction**
1.	User selects an existing influencer profile	System loads profile details.
2.	User modifies handles or metadata	System validates updated data.
3.	User submits changes	System updates profile & confirms success.

Alternate Scenarios:

2a: Invalid handle format → Highlight invalid fields.
3a: Update conflicts with existing profile → Show duplication warning.

Post Conditions

Step#	**Description**
—	Profile information is updated in the system.

Use Case Cross referenced	**UC-PROFILE-01: Add Profile** **UC-SCAN-01: Run Scan**

UC-PROFILE-03: Delete Influencer Profile

Use case Id:	UC-PROFILE-03
Actors:	Brand Manager / HR / Agency User
Feature:	Profile Removal
Pre-condition:	Profile exists and user has deletion rights

Scenarios

Step#	**Action**	**Software Reaction**
1.	User selects a profile to delete	System shows confirmation dialog.
2.	User confirms deletion	System deletes profile & updates list.

Alternate Scenarios:

1a: Profile linked to active scans → System prevents deletion.
2a: Server issue → Show *"Deletion failed."*

Post Conditions

Step#	**Description**
—	Profile removed from system.

Use Case Cross referenced	**UC-SCAN-01: Run Scan**

UC-SCAN-01: Initiate Scan

Use case Id:	UC-SCAN-01	
Actors:	Brand Manager / HR / Agency User	
Feature:	Social Media Behavior Scanning	
Pre-condition:	Influencer profile is added	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User selects “Run Scan”	System loads scan configuration.
2.	User clicks Start Scan	System begins scraping content.
3.	System analyzes posts/tweets	Shows progress bar.
4.	System generates risk score	Shows results dashboard.
Alternate Scenarios:		
2a: Social API rate limit → System queues scan. **3a: Private account detected midway → Scan aborted.**		
Post Conditions		
Step#	**Description**	
—	Scan results stored in database.	
Use Case Cross referenced	**UC-SCAN-02: View Scan Results**	

UC-SCAN-02: View Scan Results

Use case Id:	UC-SCAN-02	
Actors:	All system users except guests	
Feature:	Result Viewing	
Pre-condition:	At least one scan completed	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User opens Results Dashboard	System loads scan history.
2.	User selects a report	System displays flagged posts and risk metrics.
3.	User downloads report (PDF/CSV)	System generates downloadable file.
Alternate Scenarios:		
1a: No scans exist → Show *“No results available.”* **3a: File generation error → Retry prompt.**		
Post Conditions		
Step#	**Description**	
—	User successfully reviews or exports scan report.	
Use Case Cross referenced	**UC-SCAN-01: Initiate Scan**	

UC-SCAN-03: Schedule Recurring Scans		
Use case Id:	UC-SCAN-03	
Actors:	Brand Manager / Agency User	
Feature:	Automated Monitoring	
Pre-condition:	Influencer profile exists	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User selects **“Schedule Scan”**	System displays scheduling options.
2.	User selects frequency (daily/weekly/monthly)	Validates input.
3.	User confirms schedule	System stores task & displays “Scheduled.”
Alternate Scenarios:		
2a: Invalid time format → Highlight error. **3a: Cron scheduler failure → Show error.**		
Post Conditions		
Step#	**Description**	
—	Recurring task created in scheduler.	
Use Case Cross referenced	**UC-SCAN-01: Initiate Scan**	

UC-SCAN-04: Cancel Scan		
Use case Id:	UC-SCAN-04	
Actors:	Brand Manager / Agency User	
Feature:	Scan Termination	
Pre-condition:	Scan must be running or scheduled	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User selects a running/scheduled scan	Shows cancel option.
2.	User clicks Cancel	System terminates or deletes schedule.
Alternate Scenarios:		
2a: Scan already completed → Show *“Already finished.”* **2b: Scan engine unresponsive → System retries cancellation.**		
Post Conditions		
Step#	**Description**	
—	Scan terminated or schedule removed.	
Use Case Cross referenced	**UC-SCAN-01: Initiate Scan**	

<table>
<tr><th colspan="3">UC-SCAN-05: View Scan Progress</th></tr>
<tr><td colspan="2"><b>Use case Id:</b></td><td>UC-SCAN-05</td></tr>
<tr><td colspan="2"><b>Actors:</b></td><td>Brand Manager / HR / Agency User</td></tr>
<tr><td colspan="2"><b>Feature:</b></td><td>Progress Monitoring</td></tr>
<tr><td colspan="2"><b>Pre-condition:</b></td><td>Scan must be in progress</td></tr>
<tr><td colspan="3"><b>Scenarios</b></td></tr>
<tr><td><b>Step#</b></td><td><b>Action</b></td><td><b>Software Reaction</b></td></tr>
<tr><td>1.</td><td>User opens progress panel</td><td>System fetches real-time status.</td></tr>
<tr><td>2.</td><td>User waits for updates</td><td>Progress bar updates with % and ETA.</td></tr>
<tr><td colspan="3"><b>Alternate Scenarios:</b></td></tr>
<tr><td colspan="3"><b>1a: Connection lost → Show <i>“Unable to load progress.”</i></b>
<b>2a: Scan stalls → System notifies user and auto-restarts process.</b></td></tr>
<tr><td colspan="3"><b>Post Conditions</b></td></tr>
<tr><td><b>Step#</b></td><td colspan="2"><b>Description</b></td></tr>
<tr><td>—</td><td colspan="2">User informed about scan completion, failure, or updates.</td></tr>
<tr><td colspan="2"><b>Use Case Cross referenced</b></td><td><b>UC-SCAN-01: Initiate Scan</b></td></tr>
</table>

<table>
<tr><th colspan="3">UC-ADMIN-02: View Audit Logs</th></tr>
<tr><td colspan="2"><b>Use case Id:</b></td><td>UC-ADMIN-02</td></tr>
<tr><td colspan="2"><b>Actors:</b></td><td>Admin</td></tr>
<tr><td colspan="2"><b>Feature:</b></td><td>System Auditing</td></tr>
<tr><td colspan="2"><b>Pre-condition:</b></td><td>Admin must be logged in</td></tr>
<tr><td colspan="3"><b>Scenarios</b></td></tr>
<tr><td><b>Step#</b></td><td><b>Action</b></td><td><b>Software Reaction</b></td></tr>
<tr><td>1.</td><td>Admin opens Audit Logs</td><td>System loads logs (Login, Scans, Changes).</td></tr>
<tr><td>2.</td><td>Admin filters by user/type/date</td><td>System applies filters instantly.</td></tr>
<tr><td>3.</td><td>Admin exports logs</td><td>System generates CSV/PDF.</td></tr>
<tr><td colspan="3"><b>Alternate Scenarios:</b></td></tr>
<tr><td colspan="3"><b>1a: No logs → Display empty state.</b>
<b>3a: Export error → Retry or fallback.</b></td></tr>
<tr><td colspan="3"><b>Post Conditions</b></td></tr>
<tr><td><b>Step#</b></td><td colspan="2"><b>Description</b></td></tr>
<tr><td>—</td><td colspan="2">Audit data accessed or exported.</td></tr>
<tr><td colspan="2"><b>Use Case Cross referenced</b></td><td><b>UC-ADMIN-01: Manage Users</b></td></tr>
</table>

5. Non-functional Requirements

5.1. Performance Requirements

Response Time:

- **Standard API Requests**: 95% of requests should complete within 2 seconds
- **Scan Initiation**: Scan job creation should complete within 3 seconds
- **Dashboard Loading**: Initial dashboard load should complete within 3 seconds
- **Scan Processing Time**:
  - Small profile (< 50 posts): 5-10 minutes
  - Medium profile (50-200 posts): 10-30 minutes
  - Large profile (200+ posts): 30-60 minutes
- **Report Generation**: PDF generation should complete within 15-30 seconds

Throughput:

- System should support at least **50 concurrent users** without performance degradation
- System should handle **20 concurrent scan jobs** (limited by AI model inference capacity)
- Database should handle **1000 queries per second** under normal load

Scalability:

- System should be horizontally scalable (add more server instances as user base grows)
- Database should support sharding for handling large data volumes (> 1TB)
- AI inference services should support GPU auto-scaling for peak demand

Resource Utilization:

- **CPU Usage**: Average < 70% under normal load, < 90% during peak
- **Memory Usage**: Average < 80% of available RAM
- **Database Connections**: Connection pool should efficiently manage connections (max 100 connections)
- **API Rate Limits**: External APIs should not be exhausted (stay within platform limits)

Reliability:

- **System Uptime**: 99.9% uptime (< 8.76 hours downtime per year)
- **Data Backup**: Daily automated backups with 30-day retention
- **Disaster Recovery**: Recovery Time Objective (RTO) < 4 hours, Recovery Point Objective (RPO) < 24 hours

5.2. Safety Requirements

Data Integrity:

- All database transactions must be ACID-compliant (Atomicity, Consistency, Isolation, Durability)
- Critical data updates (user credentials, scan results) must be logged for audit trails
- System should prevent data corruption through validation checks and error handling

Error Handling:

- System should gracefully handle errors without crashing
- User-facing errors should display helpful messages (no technical jargon or stack traces)
- Critical errors should be logged and escalated to administrators
- Failed scan jobs should be retried up to 3 times before marking as failed

Failover Mechanisms:

- Database replication (primary-replica setup) to prevent data loss
- Load balancer health checks to route traffic away from unhealthy servers
- Queue-based job processing to prevent job loss during service restarts

Safe Deletion:

- Soft delete for user accounts and profiles (mark as deleted, retain for 30 days)
- Confirmation dialogs for destructive actions (delete profile, delete user)
- Admin actions should require additional authentication (e.g., re-enter password)

AI Model Safety:

- Models should be evaluated for fairness and bias before deployment
- Models should not output harmful content or recommendations
- Confidence thresholds should prevent false positives from causing undue harm to influencers' reputations

5.3. Security Requirements

Authentication & Authorization:

- **Multi-Factor Authentication (MFA)**: Recommended for admin accounts (optional for standard users)
- **JWT Token Authentication**: Stateless authentication with 24-hour token expiration
- **Role-Based Access Control (RBAC)**:
    - **Admin**: Full access to all features, user management, system settings
    - **Brand Manager**: Create scans, view results, generate reports for own organization
    - **HR Professional**: Same as Brand Manager
    - **Agency User**: Manage multiple client profiles
    - **Parent/Guardian**: Limited access, view-only for specific profiles
- **Session Management**: Automatic logout after 30 minutes of inactivity

Data Encryption:

- **In Transit**: All data transmitted over HTTPS with TLS 1.3
- **At Rest**: Sensitive data (passwords, personal info) encrypted using AES-256
- **Password Storage**: Passwords hashed using bcrypt with salt (cost factor 12)

API Security:

- **Rate Limiting**: 100 requests per minute per user to prevent abuse
- **Input Validation**: All user inputs sanitized to prevent SQL injection, XSS, CSRF attacks
- **CORS Policy**: Restrict API access to authorized domains only
- **API Key Management**: External API keys stored in secure environment variables (not in code)

Audit Logging:

- All user actions logged with timestamp, user ID, action type, IP address
- Admin actions (user creation, deletion, system changes) logged with additional details
- Logs stored securely and retained for 1 year

Privacy Compliance:

- **Data Minimization**: Only collect data necessary for reputation analysis (no PII beyond public usernames)
- **User Consent**: Clear terms of service and privacy policy explaining data usage
- **Right to Deletion**: Users can request deletion of their data (GDPR-like principle)
- **Anonymization**: Aggregate analytics should not expose individual user behavior

Vulnerability Management:

- Regular security audits and penetration testing (at least once before production launch)
- Dependency scanning to detect vulnerable libraries (automated via CI/CD)
- Prompt patching of security vulnerabilities

5.4. User Documentation

User Manual:

- Comprehensive guide covering all system features
- Step-by-step instructions with screenshots for:
    - Account registration and login
    - Adding influencer profiles
    - Initiating and configuring scans
    - Interpreting scan results
    - Generating and downloading reports
    - Managing user accounts (admin only)
- Available as PDF download and web-based HTML version

Online Help:

- Context-sensitive help tooltips throughout the application (hover over '?' icons)
- FAQ section addressing common questions:
    - "What platforms are supported?"
    - "How accurate is the risk scoring?"
    - "What does each risk category mean?"
    - "Can I scan private accounts?"
    - "How long does a scan take?"

Tutorial Videos:

- Short video tutorials (3-5 minutes each) for key workflows:
    - Getting started with CloutCheck AI
    - Running your first scan
    - Understanding scan results
    - Generating professional reports

API Documentation (if API access is provided):

- Detailed documentation for each endpoint
- Request/response examples
- Authentication instructions
- Rate limit information
- Available via Swagger/OpenAPI specification

Troubleshooting Guide:

- Common errors and solutions:
    - Login issues (forgot password, account not verified)
    - Scan failures (invalid handle, private account, rate limit exceeded)
    - Report generation errors
- Contact information for technical support

System Requirements:

- Minimum browser versions
- Recommended internet speed
- Hardware recommendations

6. References

6.1. Research Papers:

- Ryu, J., & Han, J. (2021). "A Multidimensional Scale for Influencer Reputation"
- Peterson, A. (2025). *The Dark Side of Social Media Influencers*. Wiley
- Kim, H., Lee, S., & Park, J. (2020). "Multimodal Post Attentive Profiling for Influencer-Sponsored Advertising Posts". WSDM 2020
- Gupta, A., et al. (2024). "ToxVidLM: Multimodal Video Toxicity Detection". ACL 2024
- Ramesh, S., et al. (2024). "MuTox: Multilingual Audio Toxicity Dataset"
- Wen, L., & Zhao, Y. (2023). "Rethinking Multimodal Content Moderation from an Asymmetric Angle (AM3)"
- Zhang, Y., et al. (2023). "Multimodal Guidance Network for Missing Modality Inference"
- Silva, R., et al. (2024). "Recent Advances in Hate Speech Moderation: Multimodal Perspectives"
- Narayanan, A., et al. (2025). "Understanding and Mitigating Toxicity in Image-Text Pretraining Datasets"

6.2. Technical Documentation:

- FastAPI Documentation: https://fastapi.tiangolo.com/
- MongoDB Documentation: https://docs.mongodb.com/
- Hugging Face Transformers: https://huggingface.co/docs/transformers/
- OpenAI Whisper: https://github.com/openai/whisper
- YOLOv8 Documentation: https://docs.ultralytics.com/
- React Documentation: https://react.dev/

6.3. Social Media Platform APIs:

- Instagram Graph API: https://developers.facebook.com/docs/instagram-api/
- Twitter API v2: https://developer.twitter.com/en/docs/twitter-api
- YouTube Data API: https://developers.google.com/youtube/v3
- TikTok API: https://developers.tiktok.com/

6.4. Web Scraping Services:

- Apify Platform: https://apify.com/
- Apify Instagram Scraper: https://apify.com/apify/instagram-scraper
- Apify TikTok Scraper: https://apify.com/apify/tiktok-scraper

7. Appendices

Appendix A: System Architecture Diagram

Architecture Overview:

The CloutCheck AI system follows a microservices-oriented architecture with five primary layers:

1. **Data Ingestion Layer**: Handles content scraping from social media platforms
2. **Preprocessing & Feature Extraction Layer**: Processes raw content into AI-ready formats
3. **AI/ML Model Inference Layer**: Performs multimodal risk classification
4. **Multimodal Fusion & Scoring Layer**: Combines insights and calculates reputation scores
5. **Application & Presentation Layer**: User-facing interfaces and reporting

Appendix B: Database Schema

Primary Collections/Tables:

1. **Users**: User accounts and authentication
2. **InfluencerProfiles**: Influencer profiles with social media handles
3. **ScanJobs**: Scan metadata and status tracking
4. **ContentItems**: Scraped content (text, images, videos, audio)
5. **RiskAnalysis**: AI model outputs and risk scores
6. **Reports**: Generated PDF reports
7. **AuditLogs**: System and user action logs

[Note: Detailed schema provided in Software Design Specification document]

Appendix C: AI Models Specification

Text Analysis Models:

- **Model**: RoBERTa-base fine-tuned on HateXplain dataset
- **Task**: Hate speech and toxicity detection
- **Input**: Tokenized text (max 512 tokens)
- **Output**: Multi-label classification probabilities

Image Analysis Models:

- **Model**: YOLOv8 for object detection
- **Task**: NSFW, violence, drugs, weapons detection
- **Input**: Image (resized to 640x640)
- **Output**: Bounding boxes, class labels, confidence scores

Audio Analysis Models:

- **Model**: OpenAI Whisper (base model)
- **Task**: Speech-to-text transcription
- **Input**: Audio file (WAV, MP3, M4A)

- **Output**: Transcribed text with timestamps

Appendix D: Sample API Endpoints

Authentication Endpoints:

- POST /api/auth/register: Register new user account
- POST /api/auth/login: Authenticate user and receive JWT token
- POST /api/auth/logout: Invalidate user session
- POST /api/auth/reset-password: Request password reset
- POST /api/auth/verify-email: Verify email address

Profile Management Endpoints:

- GET /api/profiles: List all influencer profiles
- POST /api/profiles: Create new influencer profile
- GET /api/profiles/{profile_id}: Get profile details
- PUT /api/profiles/{profile_id}: Update profile
- DELETE /api/profiles/{profile_id}: Delete profile

Scan Endpoints:

- POST /api/scans: Initiate new scan
- GET /api/scans/{scan_id}: Get scan status and results
- DELETE /api/scans/{scan_id}: Cancel ongoing scan
- GET /api/scans: List all scans with filters
- POST /api/scans/{scan_id}/rerun: Re-run previous scan

Report Endpoints:

- POST /api/reports: Generate PDF report
- GET /api/reports/{report_id}: Get report status
- GET /api/reports/{report_id}/download: Download PDF report

Admin Endpoints:

- GET /api/admin/users: List all users (admin only)
- POST /api/admin/users: Create user (admin only)
- PUT /api/admin/users/{user_id}: Update user (admin only)
- DELETE /api/admin/users/{user_id}: Delete user (admin only)
- GET /api/admin/system-health: View system health metrics (admin only)

Appendix E: Risk Categories and Definitions

Category	Definition	Examples
Hate Speech	Content that attacks or demeans individuals/groups based on race, religion, ethnicity, gender, sexual orientation, or other protected characteristics	Racial slurs, homophobic language, sexist remarks
NSFW/Sexual	Sexually explicit or suggestive content, nudity, or adult themes inappropriate for general audiences	Explicit images, sexual innuendos, provocative poses
Violence	Depictions or promotion of physical violence, gore, weapons, or threatening behavior	Fight scenes, graphic injuries, weapon display
Drugs/Alcohol	Content showing or promoting drug use, excessive alcohol consumption, or substance abuse	Drug paraphernalia, intoxication, substance glorification
Extremism	Content promoting extremist ideologies, terrorism, or radical political views	Extremist symbols, radicalization messaging, conspiracy theories
Political Controversy	Highly divisive political statements or endorsements that may polarize audiences	Partisan attacks, controversial policy positions
Misinformation	False or misleading information that could deceive audiences	Fake news, unverified claims, conspiracy theories

Appendix F: Glossary

Term	Definition
Multimodal Analysis	Analyzing content across multiple modalities (text, images, videos, audio) simultaneously
Fusion Strategy	Technique for combining insights from different modalities into unified prediction
Risk Score	Quantitative measure (0-100) of reputation risk based on detected harmful content
Confidence Score	Probability (0-1) indicating AI model's certainty in its prediction
Flagged Content	Content identified by AI models as potentially harmful or risky
NSFW	Not Safe For Work - content inappropriate for professional settings
Toxicity	Rude, disrespectful, or offensive language intended to harm or upset
Scraping	Automated extraction of data from websites or social media platforms

Term	Definition
Embedding	Numerical vector representation of text, images, or audio for AI processing
Fine-tuning	Process of adapting pre-trained AI models to specific tasks or domains

Appendix G: Test Scenarios (Sample)

Functional Test Cases:

1. **TC-01: User Registration**
   - Input: Valid email, strong password
   - Expected: Account created, verification email sent
   - Status: [Pass/Fail]
2. **TC-02: Invalid Login**
   - Input: Incorrect password
   - Expected: Error message "Invalid email or password"
   - Status: [Pass/Fail]
3. **TC-03: Scan Initiation**
   - Input: Valid Instagram handle
   - Expected: Scan job created, status "Queued"
   - Status: [Pass/Fail]
4. **TC-04: Report Generation**
   - Input: Completed scan ID
   - Expected: PDF report generated within 30 seconds
   - Status: [Pass/Fail]

Performance Test Cases:

1. **TC-P01: Concurrent Users**
   - Load: 50 simultaneous users
   - Expected: Response time < 3 seconds
   - Status: [Pass/Fail]
2. **TC-P02: Large Scan**
   - Input: Profile with 500+ posts
   - Expected: Scan completes within 90 minutes
   - Status: [Pass/Fail]

Security Test Cases:

1. **TC-S01: SQL Injection**
   - Input: Malicious SQL in form fields
   - Expected: Input sanitized, no database breach
   - Status: [Pass/Fail]

2. **TC-S02: Unauthorized Access**
   - Input: Access admin endpoint without admin role
   - Expected: 403 Forbidden error
   - Status: [Pass/Fail]