MediaReview Social: Product Requirements Document and Technical Specification

Company Name

February 22, 2025

Contents

1 Product Requirements Document (PRD)

1.1 Executive Summary

Product Name: MediaReview Social

Purpose: A social media platform where users review and discuss movies, TV shows, and various media, enhanced by machine learning for personalized recommendations, sentiment analysis, and content moderation.

Target Audience: Media enthusiasts, critics, and casual viewers.

1.2 Product Overview

Description:
MediaReview Social combines social interactions with advanced ML-driven insights. Users can post reviews, engage in discussions, follow trends, and receive personalized media recommendations based on their preferences and review history.

Key Value Propositions:

- **Community Engagement:** Foster a community of media lovers.
- **Personalization:** Leverage ML to tailor content and recommendations.
- **Quality Insights:** Use sentiment analysis to aggregate user feedback and highlight trending topics.

1.3 Objectives & Success Metrics

Objectives:

- Launch an MVP with core review posting, user interaction, and ML-based recommendations.
- Achieve high user engagement with at least 10,000 active users in the first three months.
- Ensure scalability to support growth and real-time interactions.

Success Metrics:

- User sign-up and retention rates.
- Average time spent on the platform and review engagement.
- Accuracy and user satisfaction with ML recommendations.
- System uptime and response times.

1.4 Functional Requirements

1.4.1 User Management

- **User Registration & Authentication:** Support email/password and social media logins (e.g., Google, Facebook), password recovery, and multi-factor authentication.
- **Profile Management:** Customizable profiles (profile pictures, bio, review history) with social features (follow system, friend requests, notifications).

1.4.2 Content Creation & Interaction

- **Review Posting:** Ability to create, edit, and delete reviews with ratings (e.g., star rating, thumbs up/down) and support multimedia attachments (images, video clips).

- **Commenting & Discussions:** Threaded comments, likes, and replies on reviews; ability to share reviews on external social platforms.
- **Feed & Discovery:** Aggregated content feed with filters (recent, top-rated, trending) and search functionality by media title, genre, or user.

1.4.3 Machine Learning Features

- **Recommendation Engine:** Personalized media suggestions based on user behavior and preferences.
- **Sentiment Analysis:** Automated sentiment scoring for reviews (positive, neutral, negative).
- **Content Moderation:** Automated flagging of inappropriate content, spam detection, and alerting for human review.

1.4.4 Media & API Integration

- **External Data Sources:** Integration with media databases (e.g., IMDb, TMDb) for metadata enrichment.
- **API Integrations:** Real-time fetching of media details (cast, synopsis, trailers).

1.5 Non-Functional Requirements

- **Performance & Scalability:** Fast load times, efficient API responses, and scalable infrastructure.
- **Security & Privacy:** Secure data transmission (HTTPS), encryption, and compliance with GDPR/CCPA.
- **Usability & Accessibility:** Responsive design, intuitive UI, and adherence to WCAG standards.
- **Maintainability:** Modular codebase, thorough documentation, and automated testing (unit, integration, end-to-end).

1.6 User Stories & Use Cases

- **As a new user:** I want to sign up quickly via email or social media so that I can start posting reviews.
- **As a reviewer:** I want to post reviews with ratings and multimedia to effectively express my opinions.
- **As a reader:** I want to view personalized recommendations and trending reviews to discover new content.
- **As an admin:** I want to moderate content automatically and manually to maintain community standards.

1.7 Roadmap & Timeline (2-Month Initial Phase)

- **Weeks 1–2:** Finalize requirements, select tech stack, and design UX/UI.
- **Weeks 3–4:** Develop backend and frontend MVP; set up databases.
- **Weeks 5–6:** Integrate machine learning features and external APIs.
- **Week 7:** Beta launch, user testing, and performance/security validation.
- **Week 8:** Final bug fixes, production deployment, and post-launch monitoring.

1.8 Risks & Mitigation

- **Scalability Issues:** Early load testing and use of scalable cloud infrastructure.
- **Data Privacy Concerns:** Adherence to data protection regulations and robust security measures.
- **ML Model Accuracy:** Continuous model training with user feedback and performance monitoring.

2 Technical Specification Document

2.1 System Architecture Overview

Architecture Style: Microservices-based architecture for modularity and scalability.

Components:

- **Frontend:** Web and mobile client applications.
- **Backend:** RESTful (or GraphQL) API server handling user management, content, and ML endpoints.
- **ML Services:** Microservices for recommendation engine, sentiment analysis, and content moderation.
- **Database:** Combination of relational (e.g., PostgreSQL) and NoSQL (e.g., MongoDB) databases.
- **Integration Layer:** API gateways for external media data (IMDb, TMDb).

2.2 Technology Stack

Frontend:

- Framework: React.js or Vue.js for web; React Native or Flutter for mobile.
- State Management: Redux or Vuex.

Backend:

- Language/Framework: Node.js (Express/Koa) or Python (Django/Flask).
- API: RESTful API or GraphQL.

Machine Learning:

- Frameworks: TensorFlow, PyTorch, or scikit-learn.
- NLP Libraries: NLTK, spaCy for sentiment analysis.

Databases:

- Relational: PostgreSQL for structured data.
- NoSQL: MongoDB for unstructured data and caching.

Deployment & Infrastructure:

- Containerization: Docker.
- Orchestration: Kubernetes.
- Cloud Providers: AWS, Google Cloud, or Azure.

CI/CD:

- Tools: GitHub Actions, Jenkins, or CircleCI.

2.3 Detailed System Design

2.3.1 API Specifications

Authentication API:

- Endpoints: `/api/auth/register`, `/api/auth/login`, `/api/auth/recover`.

- Methods: POST for registration/login, GET/PUT for profile updates.

Review API:

- Endpoints: `/api/reviews`, `/api/reviews/{id}`.
- Methods: GET for fetching reviews, POST for creating reviews, PUT/DELETE for updating/deleting reviews.

User API:

- Endpoints: `/api/users/{id}`.
- Methods: GET for fetching user data, PUT for profile updates.

ML API:

- Endpoints: `/api/ml/recommendations`, `/api/ml/sentiment`.
- Methods: POST for submitting data for analysis, GET for fetching recommendations.

2.3.2 Database Schema & Data Flow

User Data Model:

- Tables/Collections for Users, Profiles, Followers, and Authentication tokens.

Review Data Model:

- Tables/Collections for Reviews, Comments, Ratings, and Media Attachments.

ML Data Pipeline:

- **Data Ingestion:** Collect user interactions and review text.
- **Processing:** Preprocess data (tokenization, normalization) for ML models.
- **Storage:** Maintain historical data for retraining models.

2.3.3 Machine Learning Pipeline

Recommendation Engine:

- **Data:** User behavior logs and review history.
- **Model:** Collaborative filtering combined with content-based filtering.
- **Deployment:** Exposed as a RESTful service.

Sentiment Analysis:

- **Data:** Text from reviews.
- **Model:** Pre-trained NLP model fine-tuned for sentiment classification.
- **Integration:** Run inference on review submission and update sentiment scores.

Content Moderation:

- **Data:** User-generated content.
- **Model:** Classifier for spam and offensive language detection.
- **Process:** Flag content and queue for human review if necessary.

2.3.4 Security & Privacy

- **Authentication & Authorization:** Secure token-based authentication (JWT) and role-based access control.
- **Data Protection:** Encryption in transit (TLS/SSL) and at rest.
- **Compliance:** Adherence to GDPR, CCPA, and other data protection laws.

2.3.5 Scalability, Performance & Monitoring

- **Scalability:** Microservices architecture with load balancers and auto-scaling groups.
- **Performance Optimization:** Caching strategies (Redis, CDN for static assets), efficient database queries, and indexing.
- **Monitoring:** Logging (ELK stack), monitoring (Prometheus, Grafana), real-time alerts, and error tracking.

2.3.6 Testing & Deployment

- **Testing:** Automated unit, integration, and end-to-end tests; performance and load testing.
- **Deployment Strategy:** CI/CD pipelines, staging environment, rollback plans, and disaster recovery.

3 Cost Estimation

- **Frontend Hosting & CDN:** $50–$100/month.
- **Backend Servers & API Services:** $100–$200/month.
- **Databases:** $50–$100/month.
- **Machine Learning Services:** $100–$300/month.
- **Additional Services (Monitoring, Logging, etc.):** $50–$100/month.