Sentiment Analysis for Amazon Product Reviews

Team Members:

Ranwah Gamal Asala

Basmala Salama

Shahd Hesham

Mona Sherif

Mennatullah Ramdan

Ibrahim Nakib

Introduction:

Phase 1: Project Planning & Management

1. Project Proposal

1.1 Overview

This project aims to develop a Sentiment Analysis System for Amazon product reviews. The system will classify customer reviews as either positive or negative based on their content. The primary goal is to provide businesses and stakeholders with actionable insights into customer satisfaction and product performance. Additionally, the project will integrate Large Language Model (LLM) APIs to perform advanced analysis, such as generating summaries and extracting key insights from reviews. Future enhancements will focus on supporting multilingual reviews to ensure scalability and broader applicability.

1.2 Objectives

- **Primary Objective:** Build a supervised machine learning model to classify Amazon product reviews as positive or negative.

- **Secondary Objectives:**

  - Integrate LLM APIs for advanced tasks such as summarization and insights generation.

  - Provide businesses with actionable insights to improve product performance and customer satisfaction.

  - Lay the groundwork for future multilingual support to handle reviews in languages other than English.

1.3 Scope

- **Data:** The project will use a preprocessed and labeled dataset of Amazon product reviews, available on Kaggle.

- **Model:** A binary classification model (e.g., Logistic Regression, Naive Bayes, or Transformer-based models like BERT) will be trained and evaluated.

- **API Integration**: The system will integrate with an LLM API to generate summaries and insights from the sentiment analysis results.

- **Future Work:** The project will explore multilingual support by leveraging translation APIs or multilingual models like mBERT or XLM-R.

2. Project Plan

2.1 Timeline (Gantt Chart)

The project will be executed over 7 weeks, with the following timeline:

Week	Tasks
Week 1-2	Data collection and preprocessing.
Week 3-4	Model training and evaluation (e.g., Logistic Regression, Naive Bayes, BERT).
Week 5	Integration with LLM API for summarization and insights generation.
Week 6	System testing and validation.
Week 7	Final report preparation and presentation.

2.2 Milestones

- **Milestone 1**: Completion of data preprocessing (End of Week 2).

- **Milestone 2:** Successful training of the sentiment classification model with acceptable accuracy (End of Week 4).

- **Milestone 3**: Integration with LLM API and generation of insights (End of Week 5).

- **Milestone 4:** Final system testing and validation (End of Week 6).

- **Milestone 5:** Submission of the final report and presentation (End of Week 7).

2.3 Deliverables

- **Preprocessed Dataset:** Cleaned and tokenized Amazon product reviews ready for model training.

- **Trained Sentiment Classification Model**: A machine learning model capable of classifying reviews as positive or negative.

- **LLM-Generated Summaries and Insights:** Concise summaries and actionable insights generated using the LLM API.

- **Final Project Report:** A comprehensive document detailing the project's methodology, results, and future work.

- Presentation: A professional presentation summarizing the project's objectives, implementation, and outcomes.

2.4 Resource Allocation

The project team consists of six members, with the following responsibilities:

Team Member	Role
Basmala Salama	Data collection and preprocessing.
Ranwah Gamal	Data preprocessing and initial model training.
Shahd Hesham	Model training and evaluation.
Mona Sherif	Model evaluation and fine-tuning.
Mennatullah Ramadan	API integration and insights generation.
Ibrahim Nakib	API integration and final system testing.
All Team Members	Documentation, testing, and presentation.

3. Risk Assessment & Mitigation Plan

3.1 Potential Risks

1. **Data Quality Issues:**

- Risk: Poor-quality or imbalanced data could negatively impact model performance.

- Mitigation: Perform thorough data cleaning and preprocessing. Use techniques like oversampling or undersampling to handle imbalanced data.

2**. Model Overfitting:**

- Risk: The model may perform well on training data but poorly on unseen data.

- Mitigation: Use cross-validation and regularization techniques. Evaluate the model on a separate validation set.

3. **API Integration Failures:**

- Risk: Issues with LLM API integration could delay insights generation.

- Mitigation: Test API integration early and have backup plans (e.g., alternative APIs or manual summarization).

4. **Multilingual Challenges:**

- Risk: Future multilingual support may face language-specific issues.

- Mitigation: Start with translation APIs and gradually move to multilingual models like mBERT or XLM-R.

3.2 Contingency Plans

- If data quality issues arise, the team will source additional datasets or use data augmentation techniques.

- If model performance is suboptimal, alternative algorithms (e.g., SVM, Random Forest) will be explored.

- If API integration fails, the team will manually generate summaries and insights until the issue is resolved.

4. Key Performance Indicators (KPIs)

To measure the success of the project, the following KPIs will be tracked:

KPI	Target	Measurement Method
Model Accuracy	At least 85% accuracy in sentiment classification.	Evaluation on test dataset.

Response Time	Process reviews and generate insights in < 5 seconds per review.	System performance testing.
User Adoption Rate	High stakeholder adoption for decision-making.	Feedback from stakeholders.
System Uptime	99% system availability during testing.	Monitoring system uptime during testing

Phase 2: Literature Review

2.1 Feedback & Evaluation

The project will be evaluated by the course instructor based on the following criteria:

- Documentation (20%): Clarity, completeness, and organization of the project report.

- Implementation (30%): Success in building and integrating the sentiment analysis model and LLM API.

- Testing (20%): Thoroughness of testing and validation processes.

- Presentation (15%): Effectiveness of the final presentation and ability to communicate results.

- Innovation (15%): Creativity in addressing challenges and suggesting future improvements.

5.2 Suggested Improvements

- Multilingual Support: Expand the system to handle reviews in multiple languages using translation APIs or multilingual models.

- Real-Time Analysis: Implement real-time sentiment analysis for incoming reviews.

- Advanced Insights: Use more advanced LLM capabilities to generate deeper insights, such as identifying emerging trends or predicting future customer satisfaction.

5.3 Final Grading Criteria

The final grade will be determined based on the following breakdown:

Criteria	Weight
Documentation	20%
Implementation	30%
Testing	20%
Presentation	15%
Innovation	15%

Phase 3: Requirements Analysis

Stakeholder Analysis

- **Key Stakeholders**:
    - **Businesses/Product Owners**: Need insights into customer satisfaction and product performance to make data-driven decisions.
    - **Customers**: Indirectly benefit from improved products based on feedback analysis.
    - **Data Scientists/Developers**: Responsible for building, maintaining, and improving the sentiment analysis system.
    - **End-Users (e.g., Marketing Teams)**: Use the system to generate reports and insights for strategic planning.

User Stories & Use Cases

1. **Product Manager Use Case**:
    - "As a product manager, I want to see sentiment trends over time so I can track the impact of product changes."
    - "As a product manager, I want to identify specific product features mentioned in negative reviews so I can prioritize improvements."
2. **Marketing Team Use Case**:
    - "As a marketing specialist, I want to extract positive sentiment highlights to use in promotional materials."
    - "As a marketing manager, I want to compare sentiment across different product categories to optimize marketing spend."
3. **Customer Support Use Case**:
    - "As a support team lead, I want to quickly identify emerging negative sentiment patterns to proactively address customer concerns."

- "As a support agent, I want summaries of common issues to prepare appropriate responses."

4. **Business Executive Use Case**:
    - "As an executive, I want to see overall sentiment metrics across our product lines."
    - "As a business leader, I want actionable insights generated from review sentiment to inform strategic decisions."

Functional Requirements

1. **Data Collection and Management**:
    - System must import Amazon product reviews from Kaggle dataset.
    - System must preprocess text data (remove stop words, handle special characters, tokenize).
    - System must store processed reviews in an organized database.
2. **Sentiment Analysis**:
    - System must classify reviews as positive or negative with at least 85% accuracy.
    - System must provide confidence scores for each classification.
    - System must identify key phrases contributing to sentiment classification.
3. **LLM Integration**:
    - System must integrate with LLM APIs for advanced text analysis.
    - System must generate concise summaries of sentiment trends.
    - System must extract actionable insights from aggregated review data.
    - System must correlate sentiment with specific product features or aspects.
4. **User Interface**:
    - System must provide a simple Streamlit-based web interface for review submission and analysis.
    - System must include a file upload component that accepts JSON and CSV formats.
    - System must validate uploaded files for proper format and display appropriate error messages.

- System must display classification results (positive/negative) with confidence score.
- System must show a brief sentiment report with key phrases highlighted.

Non-functional Requirements

1. **Performance**:
   - System must analyze and classify new reviews within 5 seconds.
   - Interface must respond to user input within 2 seconds.
2. **Security**:
   - System must implement secure API authentication for LLM integration.
   - System must not store personally identifiable information from submitted reviews.
3. **Usability**:
   - Interface must be intuitive and require no training to use.
   - System should provide clear instructions for review submission.
   - Results must be presented in non-technical, easily understandable language.
4. **Reliability**:
   - System must maintain 99% uptime.
   - System must implement error handling for API failures.
5. **Scalability**:
   - Architecture must support future addition of multilingual review analysis.
   - System must be able to handle multiple concurrent users.

Phase 4: System Analysis & Design

1. Problem Statement & Objectives

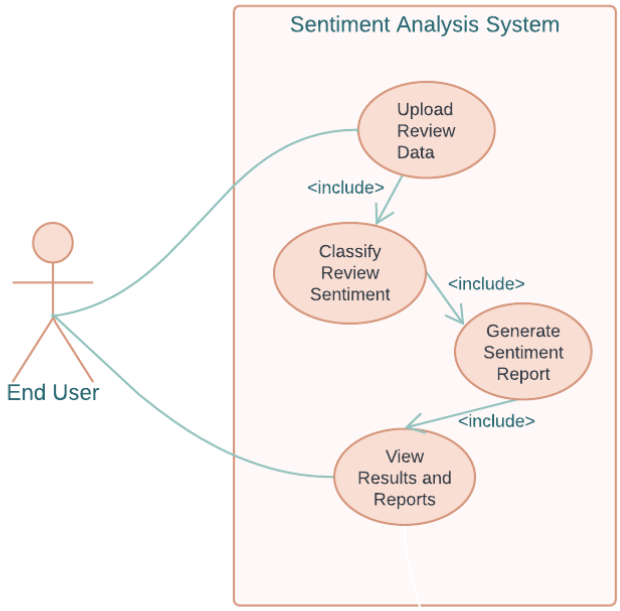
Problem Statement: Businesses and consumers need a simple way to analyze the sentiment of product reviews. Manual review analysis is subjective and time-consuming,

while existing automated solutions often lack immediate feedback and actionable insights.

Project Objectives:

1. Develop a machine learning system that accurately classifies product reviews as positive or negative.
2. Create a simple, user-friendly interface for submitting reviews and receiving immediate sentiment analysis.
3. Leverage LLM capabilities to generate meaningful insights from individual reviews.
4. Provide clear, actionable feedback that highlights key sentiment drivers in the review.
5. Establish a foundation for future expansion to multilingual sentiment analysis.

Use Case Diagram:



Primary Actors:

- End User

Key Use Cases:

1. **Upload Review Data**
    - Actor: End User

- Description: User uploads a JSON or CSV file containing product reviews to the system.
- Precondition: System is operational and user has properly formatted review data.
- Post-condition: File is uploaded and validated for processing.

2. **Classify Review Sentiment**
   - Actor: System (automated)
   - Description: System analyzes all reviews in the uploaded file and classifies sentiment as positive or negative.
   - Precondition: Valid review data file is uploaded.
   - Post-condition: All reviews are tagged with sentiment classification and confidence scores.
3. **Generate Sentiment Report**
   - Actor: System (automated)
   - Description: System creates a comprehensive report highlighting key sentiment aspects across all reviews.
   - Precondition: Reviews have been classified.
   - Post-condition: Sentiment report is generated and displayed to user.
4. **View Results and Reports**
   - Actor: End User
   - Description: User view the sentiment analysis results for future reference.
   - Precondition: Sentiment analysis is complete.
   - Post-condition: Results are downloaded in the requested format.

2. Database Design & Data Modeling

This is ERD Diagrom describe the dataset **amazon reviews for sentiment analysis**

entities

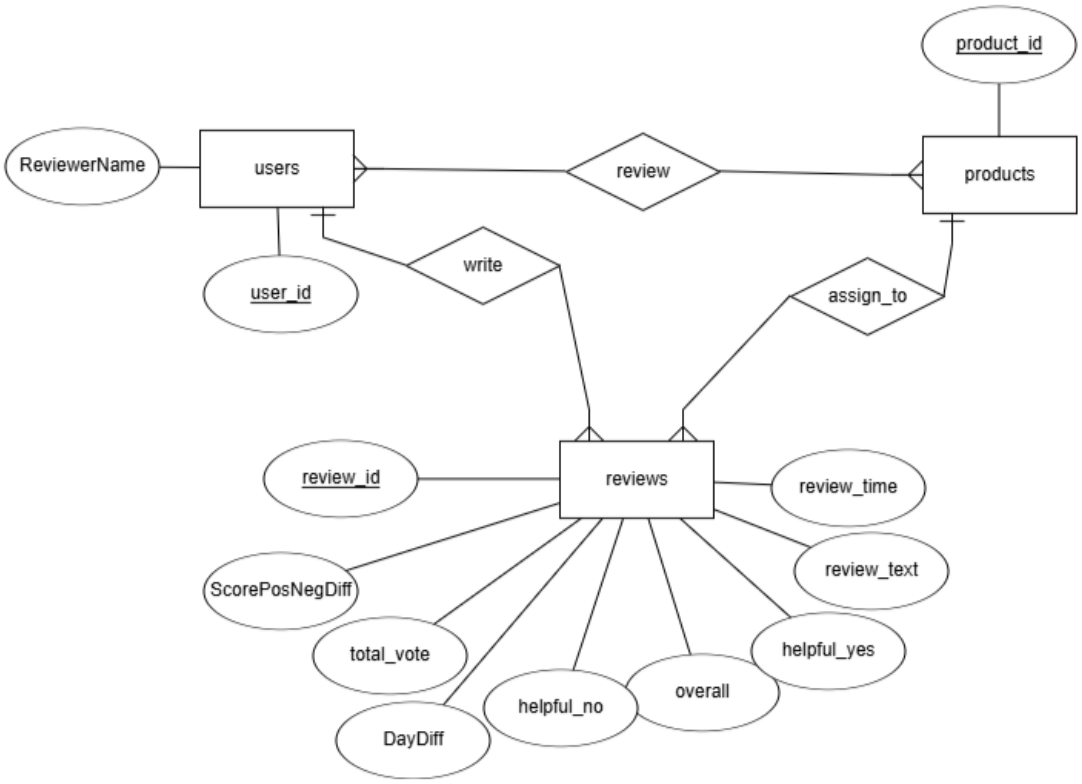
- users with attributes: user_id, ReviewerName
- products with attribute: product_id

- reviews with attributes like review_id, review_time, review_text, overall, helpful_yes, helpful_no, etc.

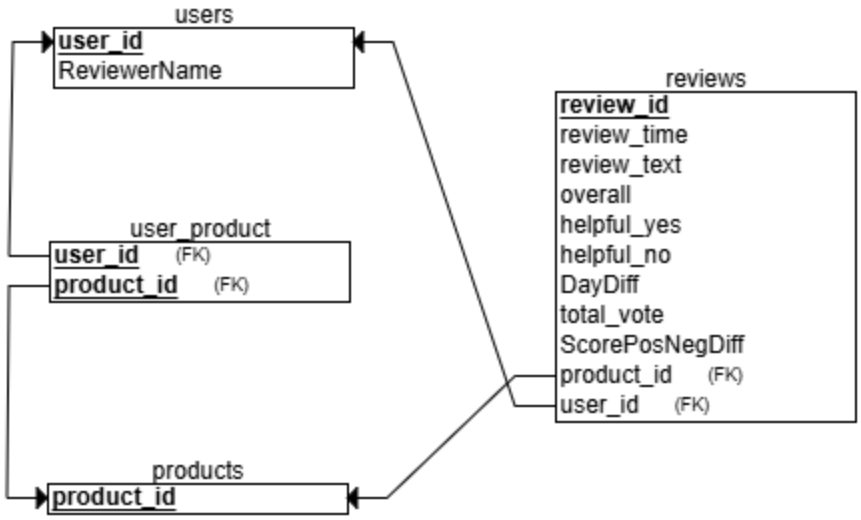
Relationships:

- Users **write** reviews (one to many)
- Reviews **are assigned to** products (one to many)
- Users **review** product (many to many)

ERD



Shema for the ERD dataset:



3. Data Flow & System Behavior

3.1. Introduction

The Data Flow & System Behavior section provides a structured overview of how data moves through the system, how different components interact, and how the system processes user inputs. Understanding these workflows ensures the system operates efficiently and meets functional requirements.

This section includes:

- **Data Flow Diagrams (DFD):** Illustrating how data is collected, processed, and stored.
- **Sequence Diagram:** Showing the interactions between users, the system, the LLM API, and the database.
- **Activity Diagram:** Representing the step-by-step flow of sentiment analysis and insights generation.
- **State Diagram:** Describing the different states a review transitions through in the system.

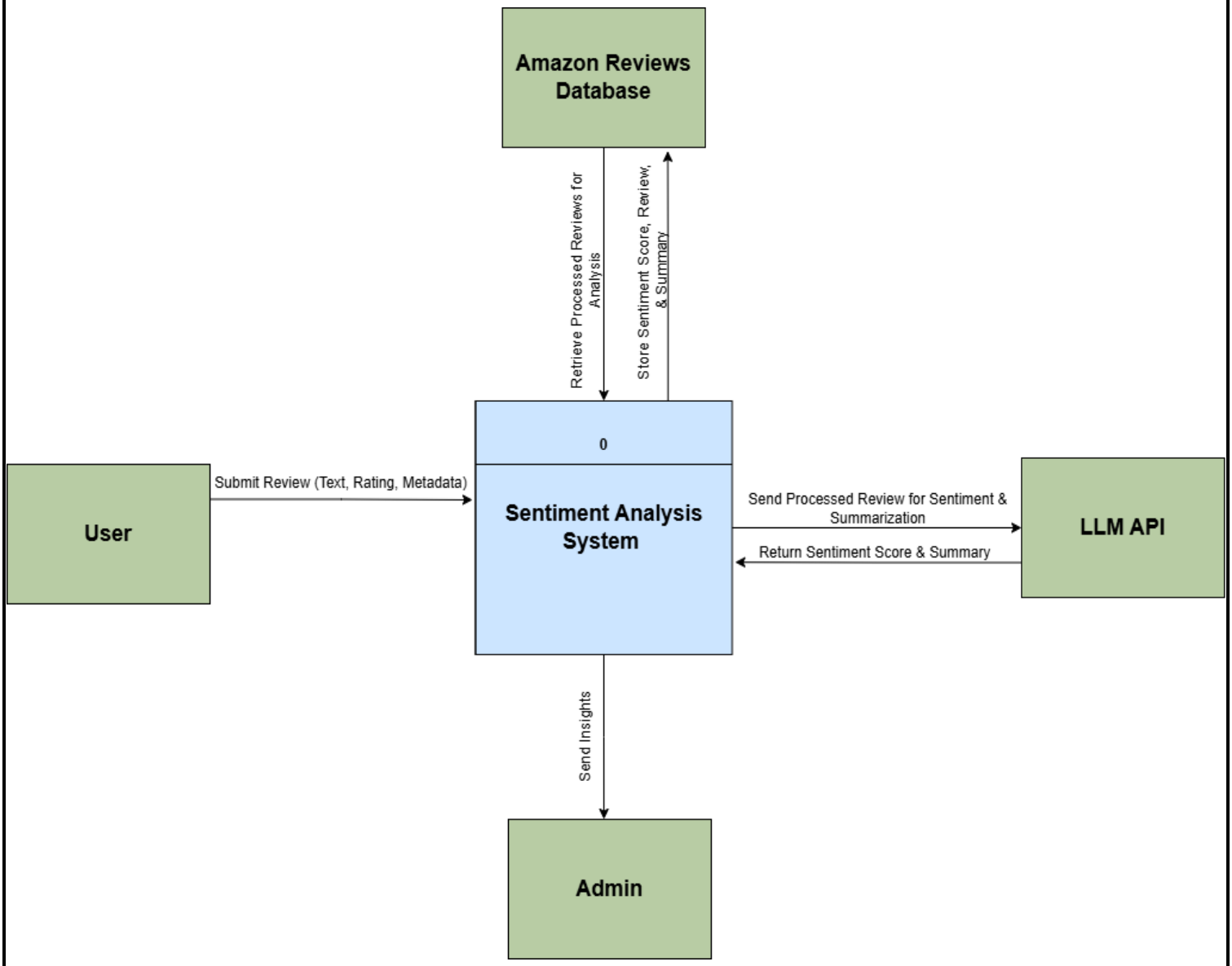
- **Class Diagram:** Defining the system's object-oriented structure, including key entities and their relationships.

These diagrams provide a visual representation of system operations, aiding developers in implementation and debugging while ensuring smooth data handling. The following subsections detail each aspect of system behavior.

3.2. Data Flow Diagram (DFD)

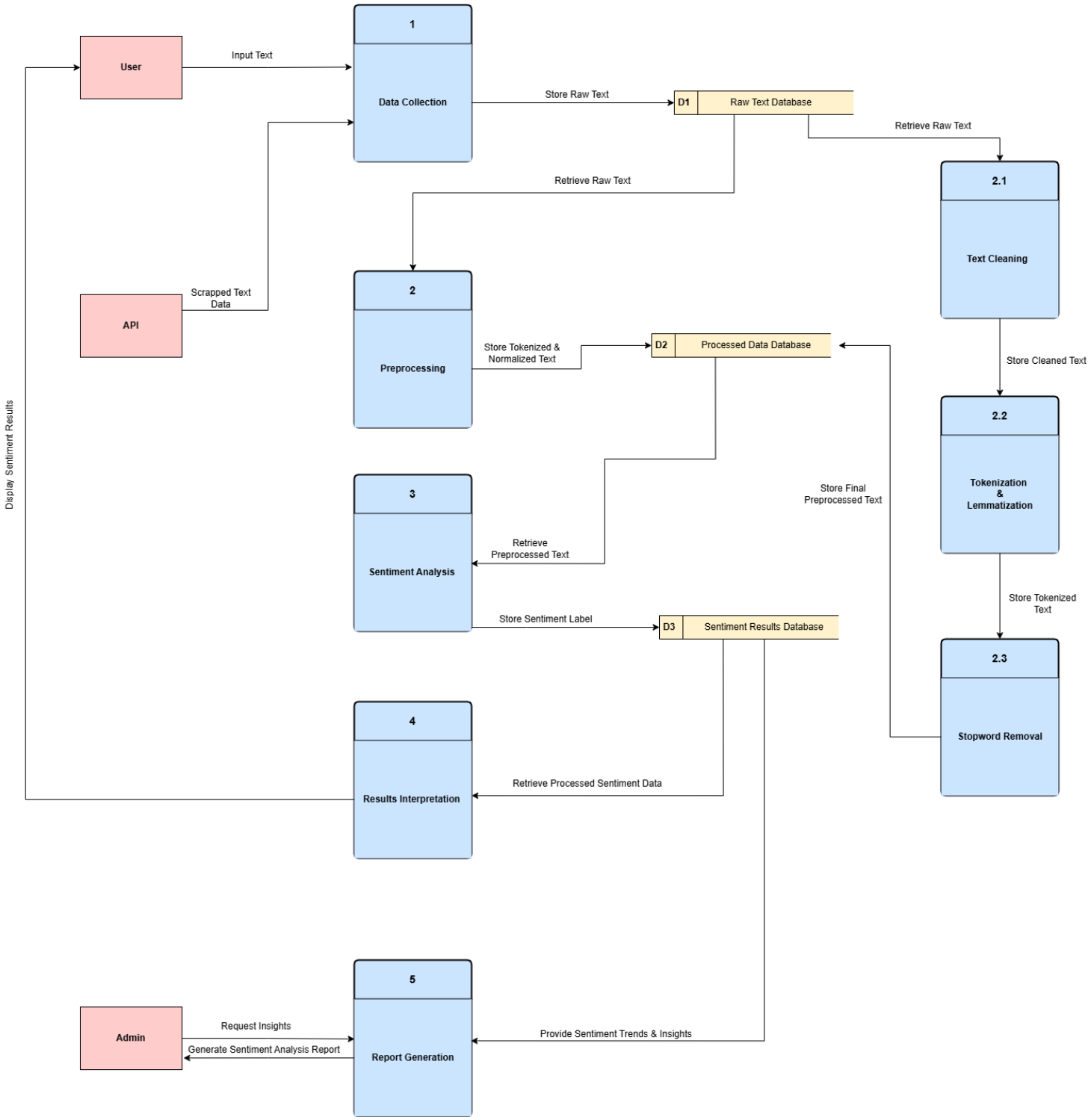
3.2.1. Context-Level DFD

Provides a high-level overview of how external entities (Users, LLM API, Database, Admin) interact with the system.

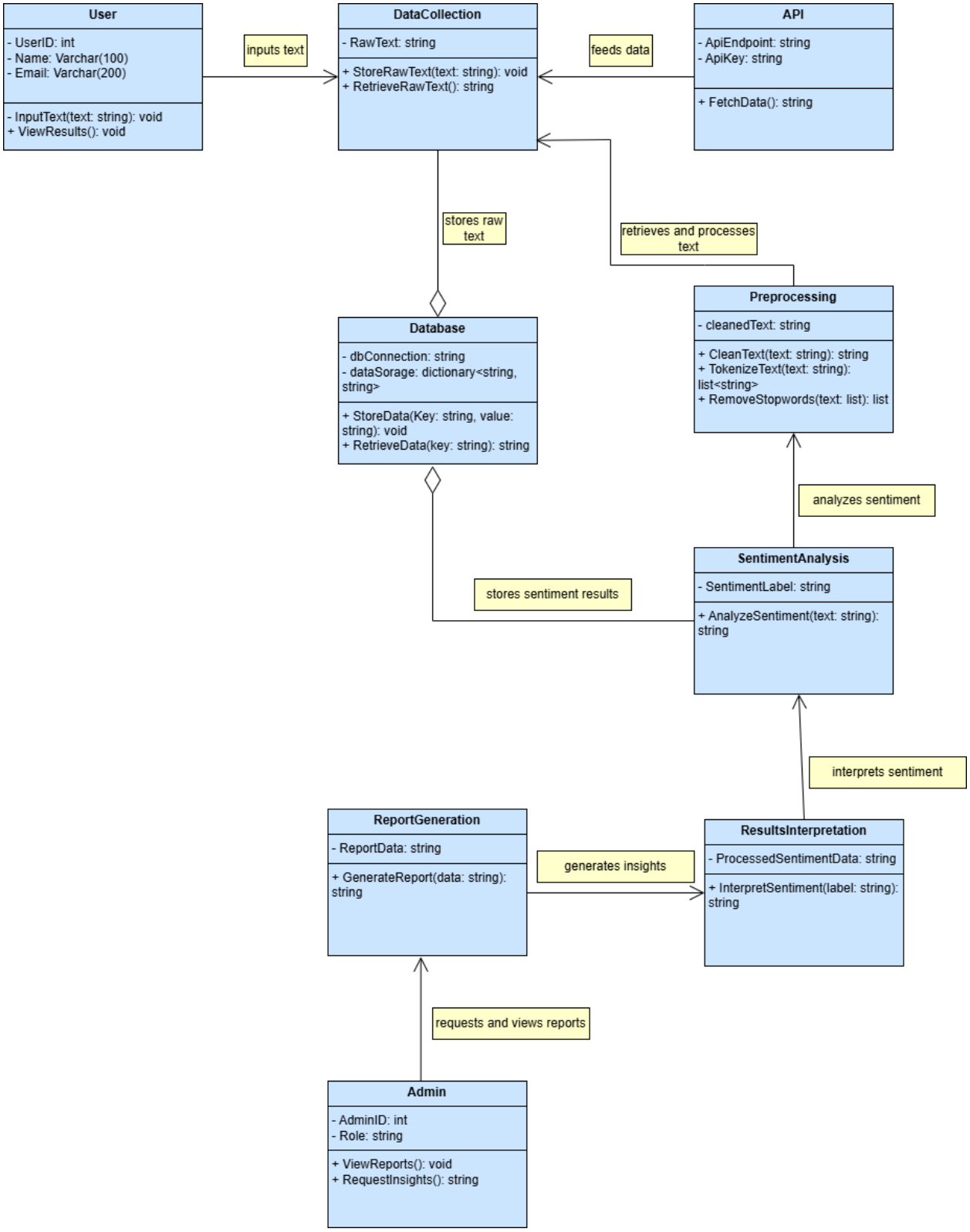


3.2.2. Level 1 DFD

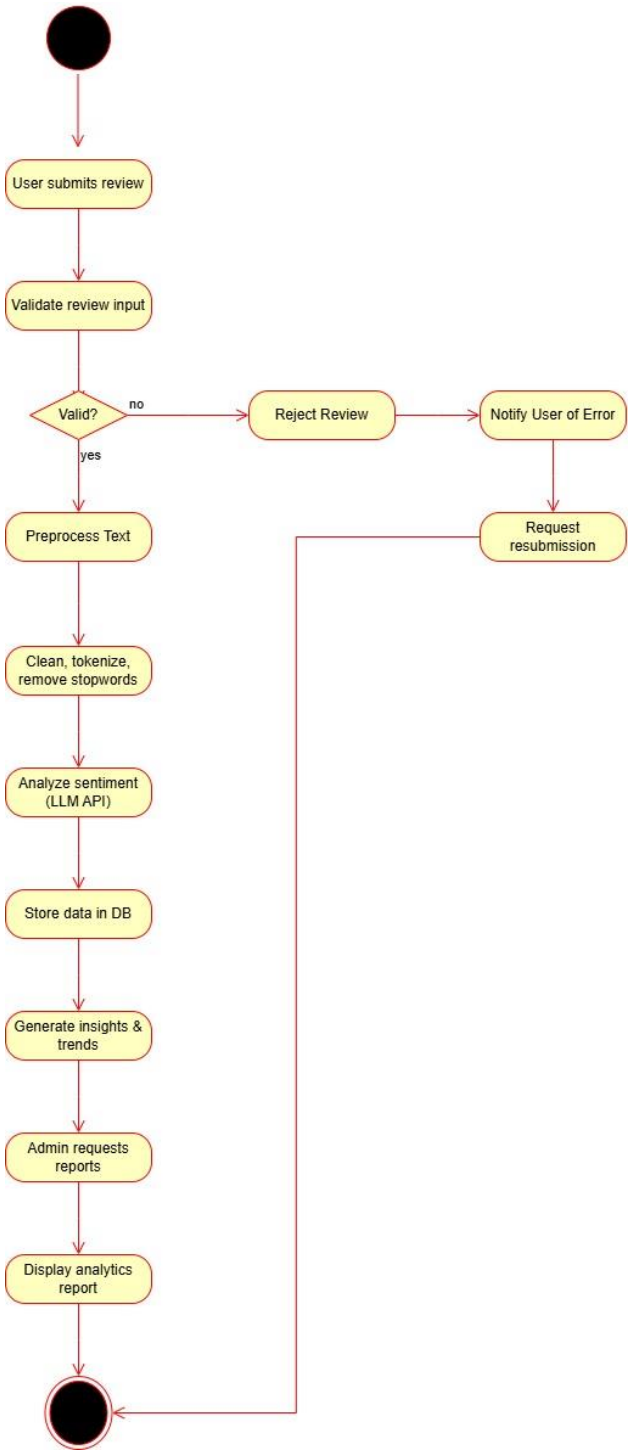
Shows detailed internal processes like review submission, preprocessing, classification, and report generation.



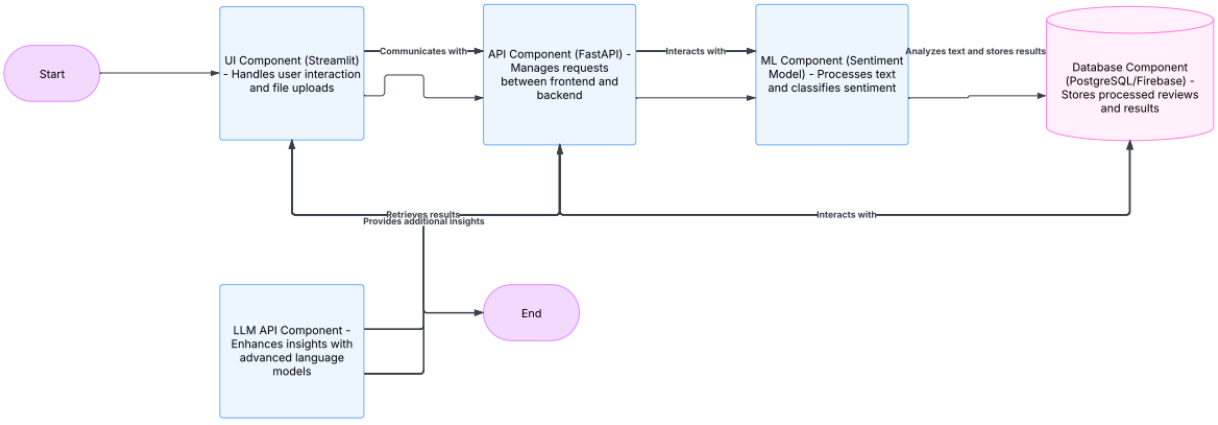
3.3. Class Diagram



3.4. Activity Diagram



3.5. Component Diagram



4. UI/UX Design & Prototyping

5. System Deployment & Integration

System Deployment & Integration 1. Technology Stack Backend: Programming Language: Python Framework: FastAPI (for handling API requests) Libraries: pandas (data processing) scikit-learn (machine learning for sentiment analysis) nltk / spaCy (natural language processing) transformers (LLM-based text analysis) Database: PostgreSQL / Firebase (for storing processed reviews) Deployment Tools: Docker, Kubernetes (for containerized deployment) Frontend: Framework: Streamlit (for a simple and interactive web interface) Technologies: HTML, CSS (if UI customization is needed) Integration: API calls to the backend for sentiment analysis Database: Primary Option: PostgreSQL (structured data storage) Alternative: Firebase Firestore (cloud-based NoSQL for scalability)