Sentiment Analysis on Social Media Using Text and Images

20211166	عبدالرحمن معتز عبدالرحمن
20211187	عمرعبدالمجيد محمد
2011004	احمد طارق عبدالرحمن
20211082	ابراهيم عاطف ابراهيم
20211182	عمر ابراهيم محمد
20211041	عمر عبدالفتاح محمد

1. Title

A Web-Based Sentiment Analysis System for Social Media Posts Utilizing Text and Images

2. Objective

This project is focused on creating a web-based application that performs sentiment analysis on social media content using both textual and visual information. By analyzing posts that include text and images or social media links, the system will classify the sentiment as positive, negative, or neutral. The goal is to develop an easy-to-use, interactive platform that can help various stakeholders analyze public sentiment on social media, offering insights into user emotions in real-time.

3. Audience and Stakeholders

Stakeholder	*Relevance*
Marketing Agencies	Help brands understand customer feedback on social media to improve strategies.
Researchers	Analyze trends in public emotions and behavior for various studies.
Political Analysts	Monitor public mood regarding political movements or events.
Social Media Companies	Improve content moderation and recommendation algorithms.
General Users	Gain insights into the emotional impact of their own posts or communities.

4. Methodology

4.1 Agile Development Approach

The project follows the **Agile Software Development Life Cycle (SDLC)** to ensure iterative progress, continuous improvement, and user-focused outcomes.
Key principles adopted:

- **Iterative Development**: Features are developed in short sprints (2–3 weeks), allowing for continuous feedback and refinement.
- **Collaboration**: Regular communication with potential users and stakeholders to gather feedback.
- **Flexibility**: Requirements and features can evolve based on feedback or changing priorities.

- **Testing & Validation**: Each sprint includes testing phases to ensure quality and reliability.

Agile benefits:

- Faster delivery of usable features.
- Quick adaptation to stakeholder feedback.
- Transparent progress tracking and workload management.

4.2 System Architecture

The system is built using a client-server architecture to ensure scalability, responsiveness, and ease of use:

- **Frontend (Client-side)**:
    - Developed using HTML, CSS, JavaScript, and Bootstrap to ensure a responsive and user-friendly interface.
    - Features:
        - Upload text manually.
        - Paste social media post or hashtag links.
        - View the sentiment analysis results with visual feedback (e.g., emoticons or colored labels).
- **Backend (Server-side)**:
    - Built with Django for robust and scalable server-side operations.
    - Hosts the sentiment analysis models: Support Vector Machine (SVM) for text and a Convolutional Neural Network (CNN) for images.
    - Handles data preprocessing, analysis, and outputs sentiment classifications.
- **Models**:
    - Text Sentiment Analysis: Uses SVM, a powerful machine learning model for text classification.
    - Image Sentiment Analysis: Uses CNN, leveraging pre-trained models like MobileNetV2 from ImageNet for deep learning-based emotion analysis from images.

4.3 Data Collection

- **Manual Input**: Users can input text or images directly or provide links to social media content.
- **Automated Collection**: Data can also be scraped from public social media platforms using tools like Selenium or APIs to fetch both textual and visual content for analysis.

4.4 Data Preprocessing

Ensures input consistency and quality:

- **Text Preprocessing**:
  - Cleaning, tokenization, stopword removal, and lemmatization.
- **Image Preprocessing**:
  - Images are resized, normalized, and augmented to ensure compatibility and robustness.

4.5 Sentiment Analysis Models

- **Text Sentiment Analysis**:
  - **Preprocessing**: Uses traditional NLP techniques.
  - **Model**: SVM trained on labeled sentiment datasets.
- **Image Sentiment Analysis**:
  - **Preprocessing**: Normalization and resizing.
  - **Model**: CNN using MobileNetV2 pretrained on ImageNet.
  - **Optimization**: Adam optimizer for adaptive learning.
  - **Fine-tuning**: Additional datasets used if necessary for emotion-specific classification.

4.6 User Interaction Flow

1. User opens the web application.
2. Inputs text or pastes a social media link.
3. System processes the input using appropriate models.
4. Sentiment is classified as positive, negative, or neutral.
5. Visual feedback is provided on-screen.

4.7 Evaluation and Testing

- **Accuracy Testing**: Validates prediction performance.
- **Response Time Testing**: Measures system speed and latency.
- **Usability Testing**: Ensures user-friendliness and accessibility.

5. Requirements

5.1 Software Requirements

Component	***Tools & Libraries***
Frontend	HTML, CSS, JavaScript, Bootstrap
Backend	Python, Django
Machine Learning	Scikit-learn (for SVM), TensorFlow (for CNN)
NLP	NLTK
Data Collection	Selenium, Social Media APIs
Visualization	Matplotlib, Seaborn

5.2 Hardware Requirements

Environment	***Specifications***
Development	Standard workstation with at least 8GB RAM and a modern CPU
Deployment	Cloud or local server with sufficient resources to handle traffic

Final Remarks

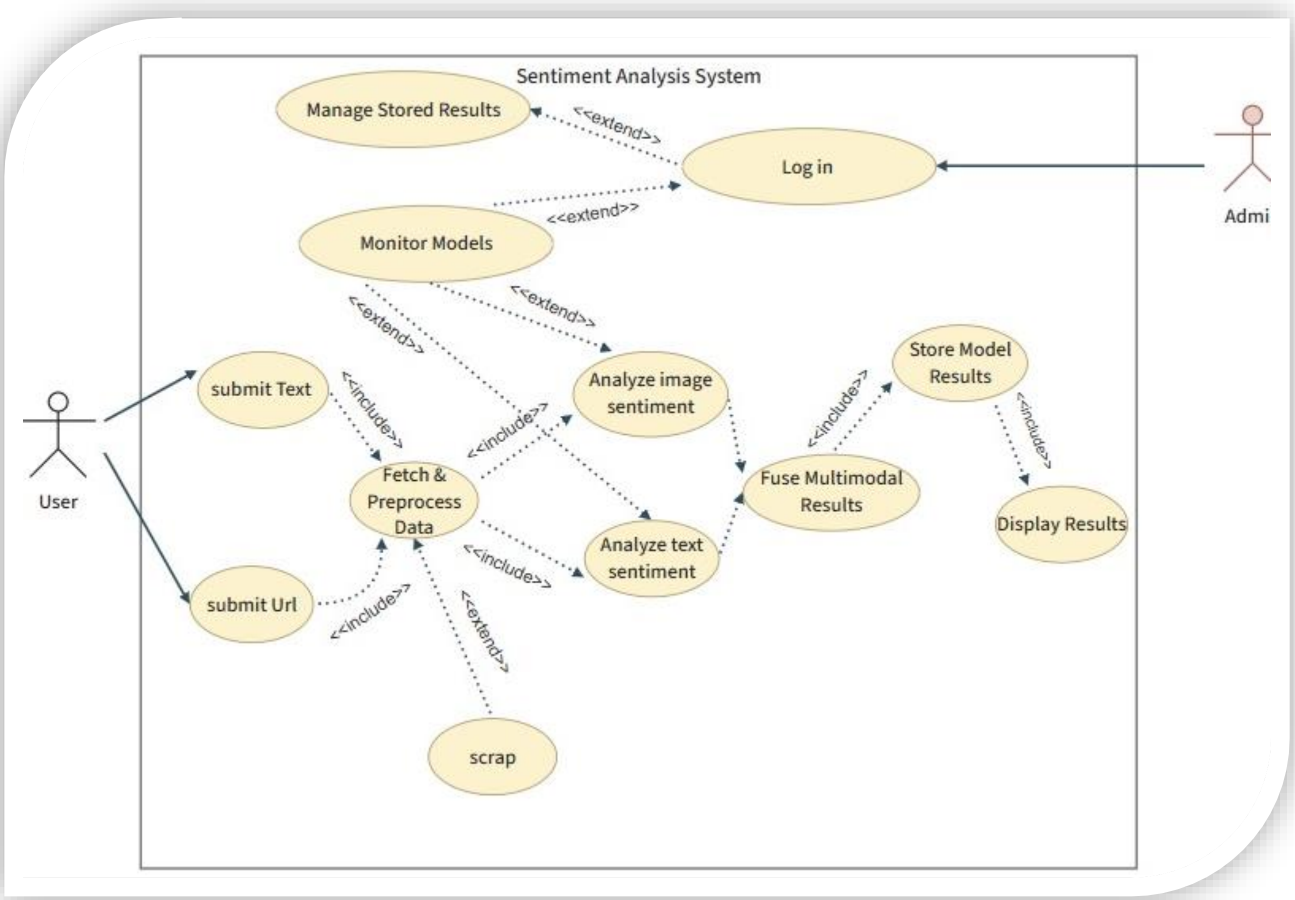
This project is the result of intensive research, development, and iteration, combining multiple disciplines such as machine learning, deep learning, natural language processing, web development, and data engineering. Developing a web-based sentiment analysis system that processes both text and images from dynamic, real-time social media content posed numerous technical and practical challenges.

Some of the key complexities we addressed include:

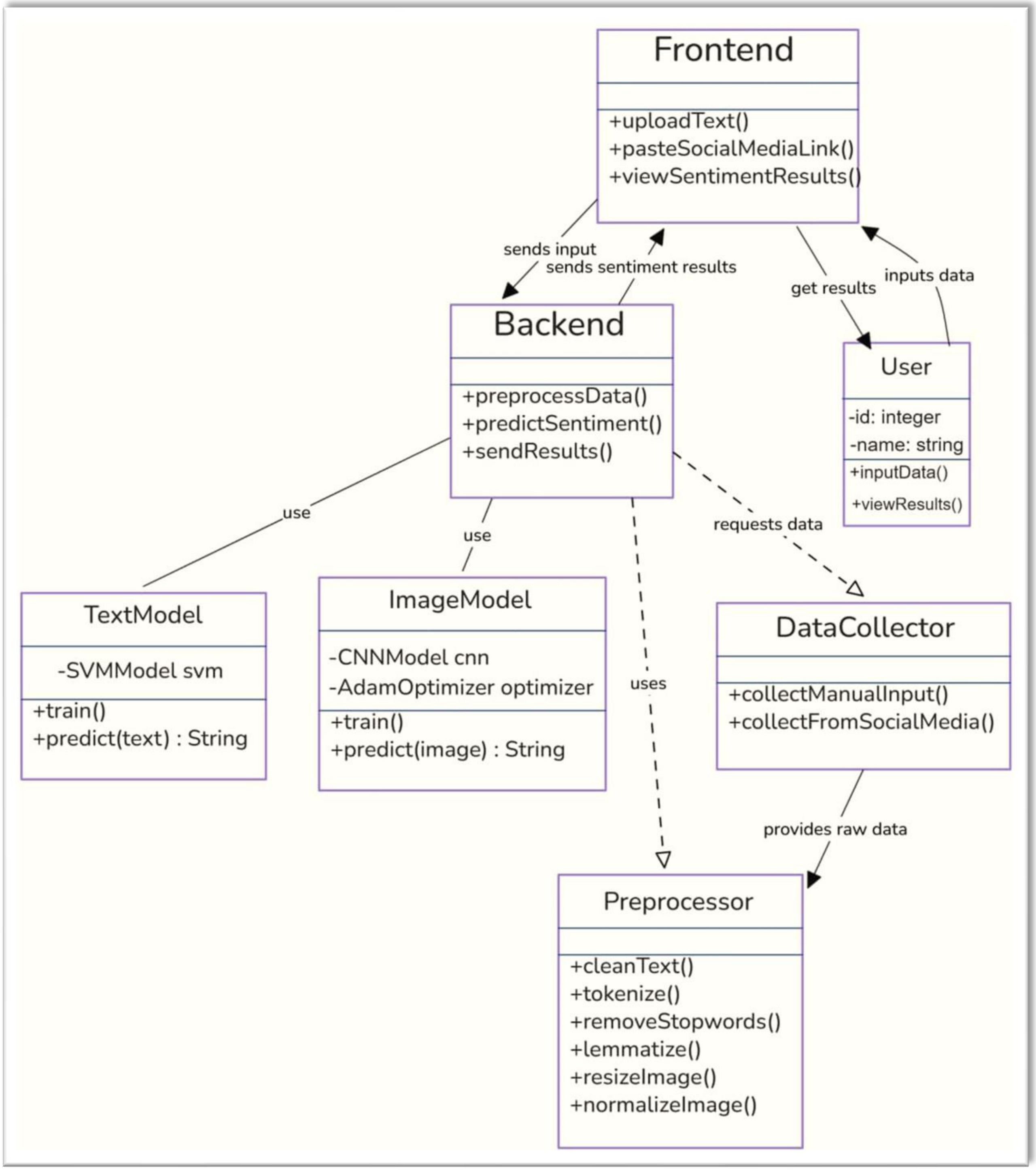
- **Dual-Modal Sentiment Analysis**: Integrating both SVM for text and CNN for images required handling two completely different data formats and building separate pipelines for preprocessing, classification, and evaluation.
- **Real-Time Social Media Data**: Social media platforms are highly dynamic and include noisy content such as emojis, slang, memes, and image filters. We engineered robust scraping mechanisms using tools like Selenium and adapted to platform restrictions, rate limits, and varied content formats.
- **Text Cleaning and Filtering**: Removing spam, emojis, and irrelevant content, along with accurate lemmatization, required customized NLP pipelines that could handle informal and diverse language styles commonly found on social media.
- **Image-Based Emotion Detection**: Social media images are often unstructured and lack clear emotional indicators. Our CNN model had to be fine-tuned to detect subtle emotional cues and work effectively with varied image types.
- **Time and Resource Management**: Building a system with multiple integrated components under tight academic deadlines demanded effective sprint planning, collaborative teamwork, and persistent debugging under pressure.
- **Testing and Iteration**: Each model and system component underwent rigorous validation, comparison, and refinement to ensure high accuracy, robustness, and responsiveness within the web application.

This project demonstrates not only our technical capabilities and innovation but also our dedication, resilience, and problem-solving skills. It stands as a comprehensive and practical solution with valuable applications in fields such as marketing, politics, and social research.

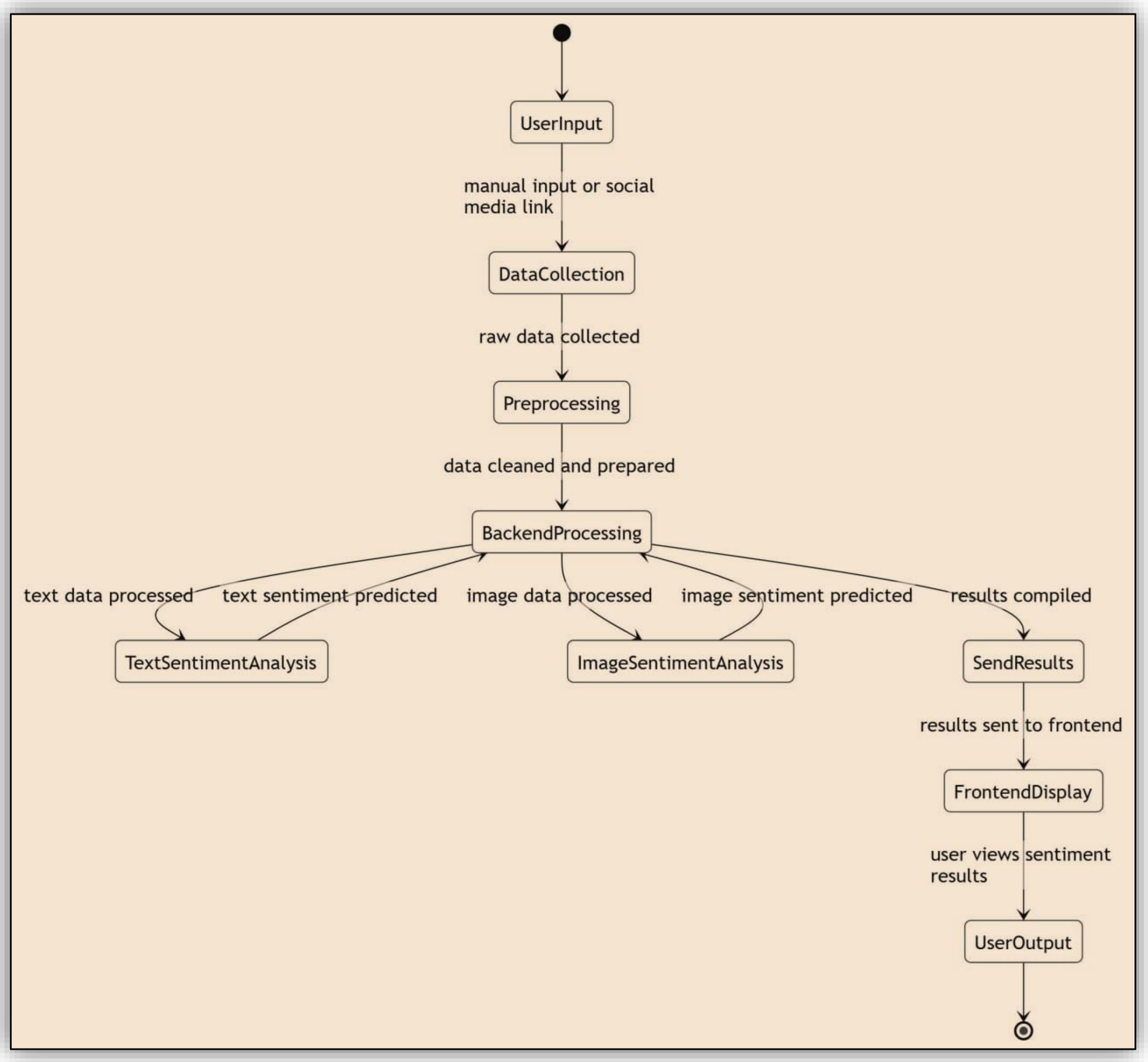
Use case diagram



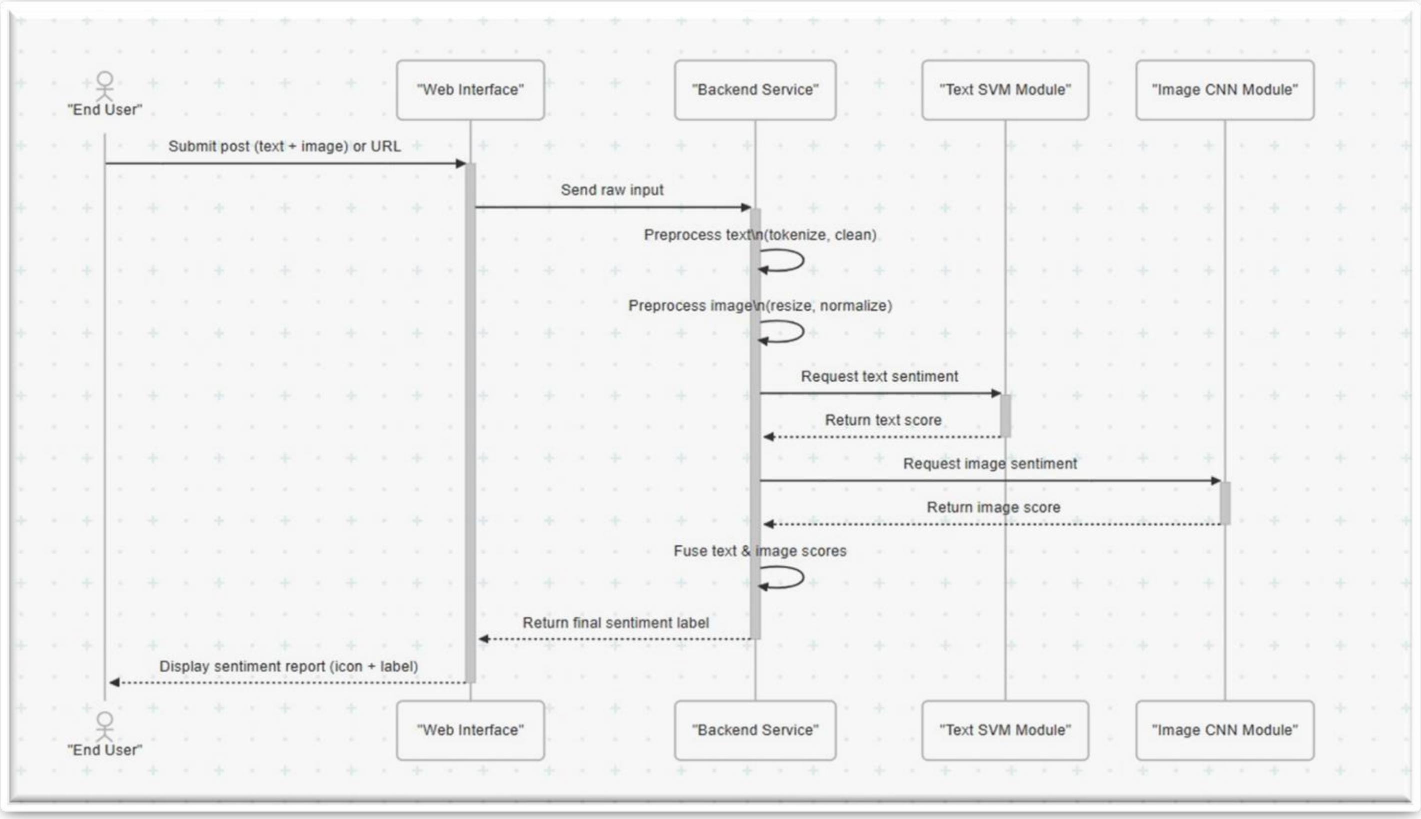
Class diagram



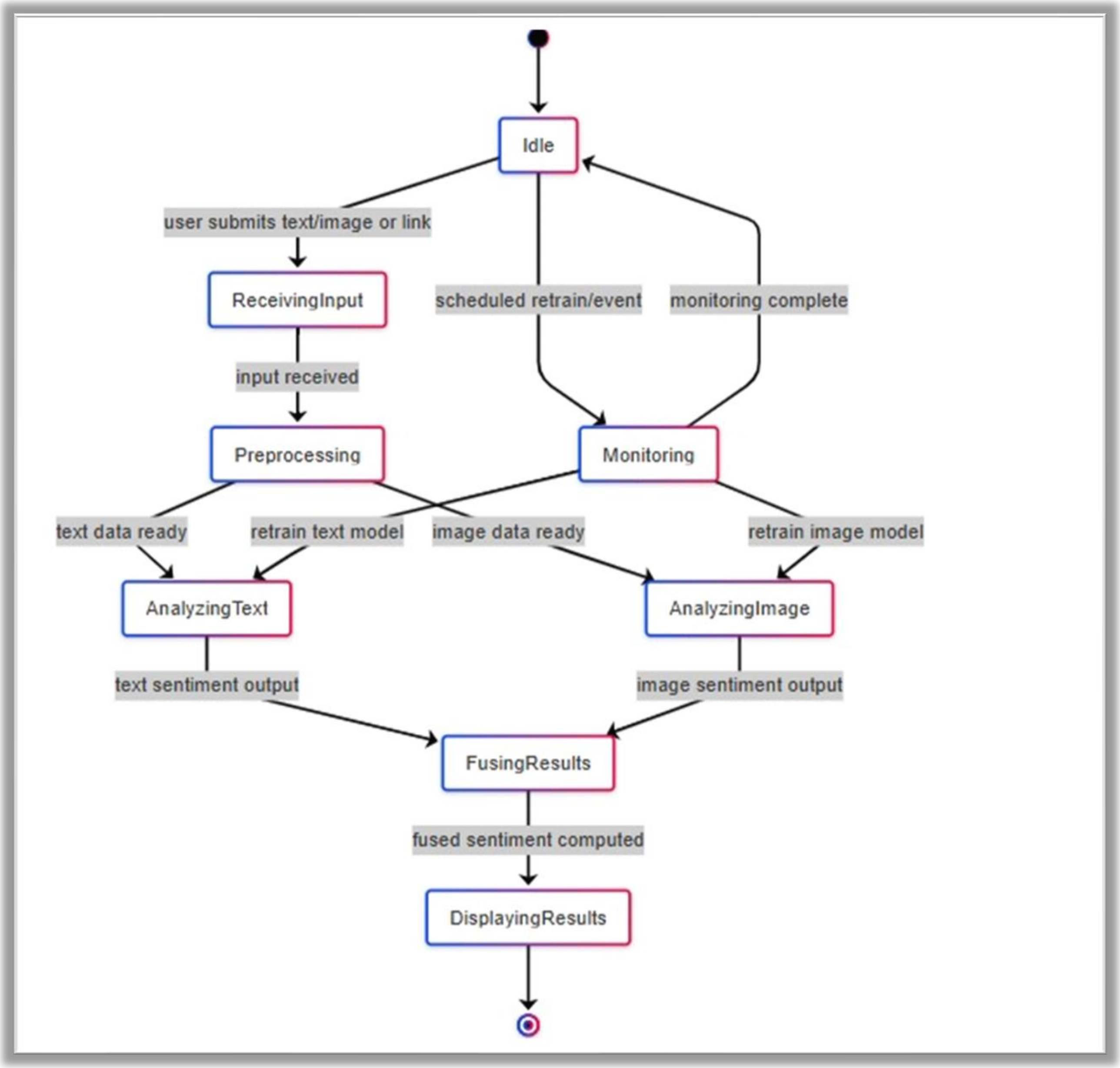
Activity diagram



sequence diagram



State diagram


Idle
user submits text/image or link
ReceivingInput
scheduled retrain/event
monitoring complete
input received
Preprocessing
Monitoring
text data ready
retrain text model
image data ready
retrain image model
AnalyzingText
AnalyzingImage
text sentiment output
image sentiment output
FusingResults
fused sentiment computed
DisplayingResults