| High Level Design Document  **Indexing and Summarization of Sports**  **Videos using Multi-Modal Approach**  UE21CS390A – Project Phase – 1  ***Submitted by:***   | Krupashree MV  Meenal Bagare  Melvin Jojee Joseph  Naveen Reddy G | PES2UG21CS242  PES2UG21CS289  PES2UG21CS294  PES2UG21CS324 | | --- | --- |   Under the guidance of   | **Dr. Sandesh B.J**  Chairperson & Professor  PES University | | --- |     **January - May 2024**  **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  FACULTY OF ENGINEERING  **PES UNIVERSITY**  (Established under Karnataka Act No. 16 of 2013)  Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India |
| --- | --- | --- | --- |

TABLE OF CONTENTS

| 1. Introduction | 3 |
| --- | --- |
| 1. Current System | 3 |
| 1. Design Considerations | 4 |
| 3.1 Design Goals | 4 |
| 3.2 Architecture Choices | 5 |
| 3.3 Constraints, Assumptions and Dependencies | 6 |
| 1. High Level System Design | 7 |
| 1. Design Description | 9 |
| * 1. Master Class Diagram | 9 |
| 5.2 Reusability Considerations | 9 |
| 1. ER Diagram / Swimlane Diagram / State Diagram | 10 |
| 1. User Interface Diagrams | 11 |
| 1. External Interfaces | 12 |
| 1. Packaging and Deployment Diagram | 13 |
| 1. Help | 14 |
|  |  |
| Appendix A: Definitions, Acronyms and Abbreviations | 15 |
| Appendix B: References | 16 |
| Appendix C: Record of Change History | 17 |
| Appendix D: Traceability Matrix | 17 |

# Introduction

This document outlines the high-level design for sports video summarization system that integrates Twitter data analysis with audio-visual processing techniques. The system aims to automatically identify significant events in sports videos based on real-time social media activity, commentator excitement, audience engagement, and scoreboard data. By leveraging cutting-edge technologies such as machine learning and natural language processing, the proposed system will deliver comprehensive and engaging sports video summaries.

# Current System [if applicable]

The current system in place for sports video summarization is primarily manual or semi-automated, relying on human editors or basic algorithms to generate summaries of sports events. This system typically involves the following components and processes:

1. **Manual Video Editing**: Human editors watch full-length sports videos and manually select key moments or highlights to include in the summary. This process is time-consuming and subjective, as it relies on individual judgment and expertise.
2. **Basic Algorithms**: In some cases, basic algorithms or rule-based systems may assist in identifying key events, such as goals, touchdowns, or significant plays. However, these algorithms often lack sophistication and may not capture the full context or importance of events accurately.
3. **Limited Integration with External Data**: The current system may have limited integration with external data sources such as social media platforms like Twitter. Any analysis of audience reactions or social media buzz is done separately from the video summarization process, leading to disjointed insights.
4. **Minimal Personalization or Customization**: Summaries generated by the current system are typically one-size-fits-all and lack personalization options for individual users. There is little consideration for varying preferences or interests among viewers.
5. **Limited Automation and Scalability**: Due to the manual nature of the process and the reliance on human editors, the current system may struggle to scale efficiently to handle a large volume of sports videos or to adapt to changing user demands and preferences.

Overall, the current system provides rudimentary summarization capabilities but lacks the sophistication, automation, and integration with external data sources necessary to deliver comprehensive and personalized sports video summaries. As such, there is a need for enhancement, replacement, or upgrade to a more advanced and automated system that can leverage cutting-edge technologies such as AI, machine learning, and real-time data analysis to deliver more engaging and insightful summaries to users.

**Design Considerations**

* 1. **Design Goals**

The design goals for the proposed system aim to address the limitations of the existing system while enhancing its capabilities and providing a more engaging and personalized user experience. These goals include:

1. **Enhanced Summarization Accuracy and Relevance**: The newly proposed system will leverage advanced algorithms and AI techniques to improve the accuracy and relevance of sports video summaries. By integrating multiple data sources, including audio, visual, and textual data from social media platforms like Twitter, the system will provide more comprehensive and insightful summaries of sports events.
2. **Improved User Experience and Interactivity**: The system will be modern, intuitive, and user-friendly, enhancing the overall user experience. The web interface will feature interactive elements and provide an intuitive experience to the users.
3. **Quality of Service**: The system will prioritize key characteristics such as availability, security, privacy, and speed to ensure a seamless and reliable user experience. It will be designed to handle high volumes of data and user requests efficiently, with minimal downtime or disruptions. Robust security measures will be implemented to protect user data and ensure privacy compliance.
4. **Social Media Insights and Engagement**: By integrating social media sources like Twitter, the system will provide users with up-to-date insights and engagement opportunities during sports events..
5. **Scalability and Adaptability**: The system will be designed to scale efficiently to handle a large volume of sports videos and user requests, adapting to changing user demands and preferences over time.

Overall, the design goals of the proposed system aim to create a more advanced, engaging, and user-centric sports video summarization platform that leverages cutting-edge technologies to deliver superior quality of service and user experience compared to the existing system.

* 1. **Architecture Choices**
* **Microservices Architecture:**
* Description: On a microservices architecture, the system is deconstructed into a set of components called microservices, each one responsible for an individual, separate function or feature. For instance, such functionality can be divided into services that can analyze Twitter data, identify replays, recognize speech, and create highlights.
* **Pros:**
* **Scalability:** A microservices architecture provides the ability to scale and flexibility by recognizing to run the services independently and in accordance with a given demand.
* **Maintainability:** The microservices architecture enables decoupling of services in the form of several components which almost separate and it assists in maintenance and clear the way for teams to modify the services independently.
* **Cons:**
* **Complexity:** Microservices architectural paradigm brings together the expending communication space among services, their deployment as well as management. Integrating such a system is not an easy task and there are a range of issues like service-to-service communication and the challenges of distributed systems.
* **Overhead:** Working on several streams all goes down to the problem of when it comes to deployment, monitoring, and coordination. Together addition of communication between services can result in higher latency and network load.
* **Sport highlighting scenarios amongst other scenarios such the microservices architecture serves the best fit. This decision is based on several factors.**
* **Scalability:** The Scientific data analysis system must manage varying loads of Twitter data and video footage capturing for replays, and gaming highlights. The use of microservices would allow each component of the system to scale separately together with demand, which will ensure that maximum performance is achieved.
* **Modularity:** Breaking system down to smaller pieces is a synonym to separation of services that serves modularity and maintainability purposes for the system. Teams can be self-sufficient to create, deploy, and enhance their services that also act as a safety valve of a complex structure and technical debt.
* **Flexibility:** In terms of services architecture, the choice of technology is essentially very flexible owing to the fact that developers can pick the best available tools and frameworks for every service. This flexibililty shall in turn trigger innovations and adaptation to the current evolving requirements.
* **Integration:** Small-batch requests in the microservices architecture design enable our company to integrate well with Twitter API, third-party services for speech recognition, and other external systems. The service itself can be formulated to address a given system integration point adaptively, becoming the key factor in achieving fully operational and reliable interrelations with the outside systems.
* The complexities as well as the overhead expenses are the drawbacks of using the microservices architectures whereas the advantages of scalability, modularity, flexibility and integration are the most essential features that made it the right choice for the developed system.

# Constraints, Assumptions and Dependencies

* **Legal Implications:**

Compliance with data protection regulations, copyright laws, and terms of service of social media platforms (such as Twitter) is crucial to avoid legal issues.

Obtaining necessary permissions and licenses for using copyrighted content (such as sports broadcasts) is essential to ensure legal compliance.

* **Usage Limitations:**

The project's success may depend on the availability and access to real-time sports data, including Twitter feeds and live video streams.

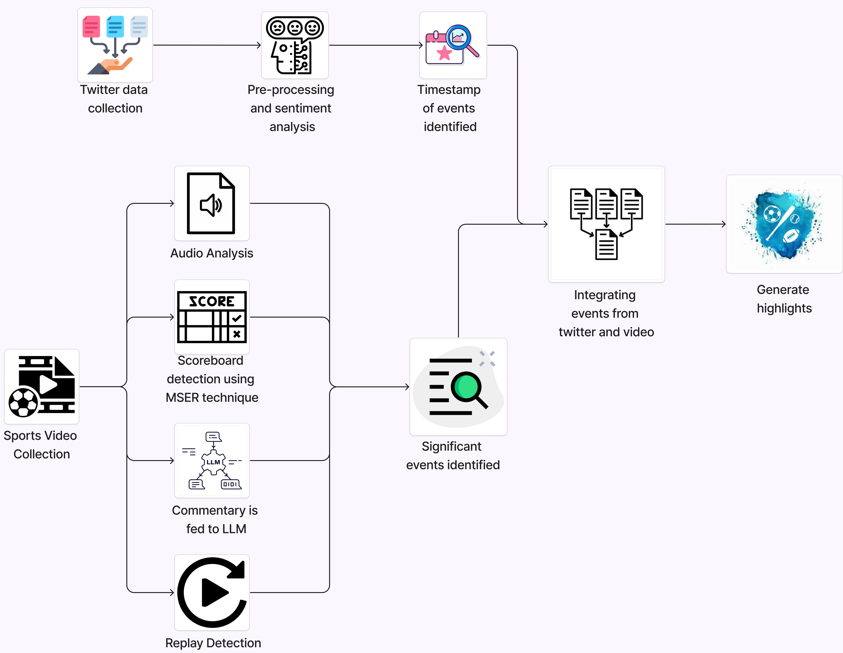
Dependence on third-party APIs or data sources may introduce usage limitations, such as rate limits or data access restrictions.

* **Assumptions Made in the Project:**

Availability of Data: Assumes the availability of sufficient and reliable data sources in real-time for analysis and summarization.

Consistency in Data Format: Assumes a level of consistency in the format and structure of data sources for effective processing.

# High Level System Design



1. **Logical User Groups:**

**System Administrators:**

Responsible for configuring and managing system components.

**Analysts:**

Utilize the system for analyzing event data and generating insights.

**Commentators/Broadcasters:**

Access commentary generated by the system for live broadcasting or analysis.

1. **Application Components:**

**Data Collection Component:**

Handles data acquisition from Twitter and video preprocessing.

**Event Detection Component:**

Includes sentiment analysis, WDHG event detection, scene detection, scoreboard analysis, and audio analysis.

**Event Linking and Analysis Component:**

Matches events detected with social media mentions, generates highlights, and integrates with LLM for commentary.

**User Interface Component:**

Provides a user-friendly interface for system interaction and visualization of analyzed data.

1. **Data Components:**

**Social Media Data:**

Tweets and social media content collected for analysis.

**Video and Audio Data:**

Game footage and audio streams processed for event detection.

**Event Data:**

Structured data representing detected events and associated metadata.

1. **Interfacing Systems:**

**External APIs:**

Interfaces with Twitter for data collection.

**LLM Service:**

Integrates with a Large Language Model service for generating commentary.

1. **Collaboration and Interaction:**

Data flows from Data Collection Component to Event Detection Component.

Event data is processed and linked by the Event Linking and Analysis Component.

Results are presented through the User Interface Component.

1. **Design Patterns and Reusability:**

**Facade Pattern:** Abstracts complex subsystems (e.g., event detection) behind simplified interfaces.

**Factory Pattern:** Used for creating different types of event detection algorithms based on data type and context.

**Singleton Pattern**: Ensures single instance of critical components like data collection manager.

**System Elements from Different Perspectives**

**1. Conceptual or Logical (UML Component Diagram):**

* **Data Collection Component:**

Twitter Data Collection Module

Video Preprocessing Module

* **Event Detection Component:**

Sentiment Analysis Module

WDHG Event Detection Module

Scene Detection Module

Scoreboard Analysis Module

Audio Analysis Module

* **Event Linking and Analysis Component:**

Event Linking Module

Highlight Generation Module

LLM Commentary Module

**2. Process - Runtime View :**

* **Sequence of Interactions:**

User interacts with the User Interface Component.

Data flows through Data Collection, Event Detection, and Event Linking Components.

Results are presented back to the User Interface for visualization.

**3. Physical - Distributed System View :**

* Nodes and Components:

Server Node hosting application components.

Client Devices accessing the User Interface Component.

External Services (Twitter, LLM Service) integrated via network interfaces.

**4. Module - Project Management and Code Organization:**

* **Project Structure:**

Separate modules for Data Collection, Event Detection, Event Linking, User Interface, and Utilities.

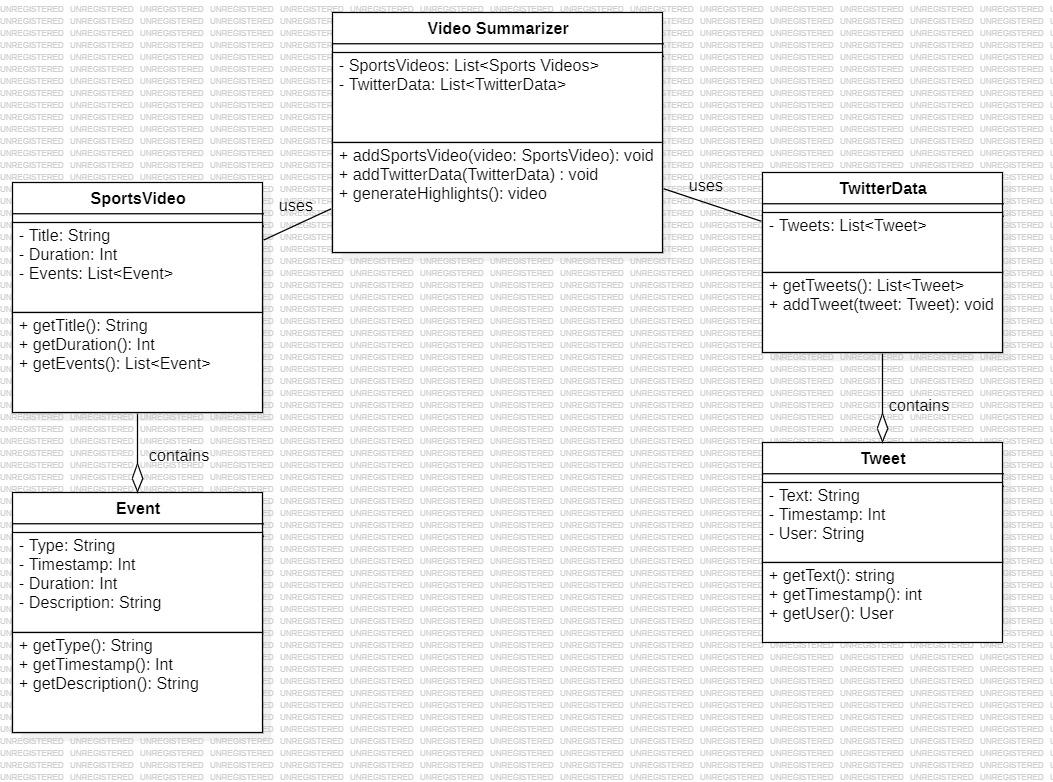
Organized codebase with clear separation of concerns and modular design for scalability and maintainability.

**5. Security - System Security Features:**

* Data Encryption: Secure data transmission and storage.
* Access Control: Role-based access control for system components.
* API Security: Secure communication with external APIs using authentication and authorization mechanisms.
* Data Privacy: Compliance with data privacy regulations and best practices.

# Design Description

* 1. **Master Class Diagram**



* 1. **Reusability Considerations**
* **Reuse of External Components:**

Utilizing established libraries and frameworks for specific functionalities such as:

Text processing (for sentiment analysis, tweet preprocessing)

Computer vision (for scene detection, scoreboard analysis)

Audio processing (for analyzing crowd noises, commentator voices)

* **APIs and Services:**

Integrate with third-party APIs and services for data collection and analysis:

Twitter API for collecting tweets and social media data

Cloud-based services for scalable audio and video processing

Leverage these APIs to reduce development effort and improve scalability.

Internal Components for Reuse:

* **Custom Modules and Utilities:**

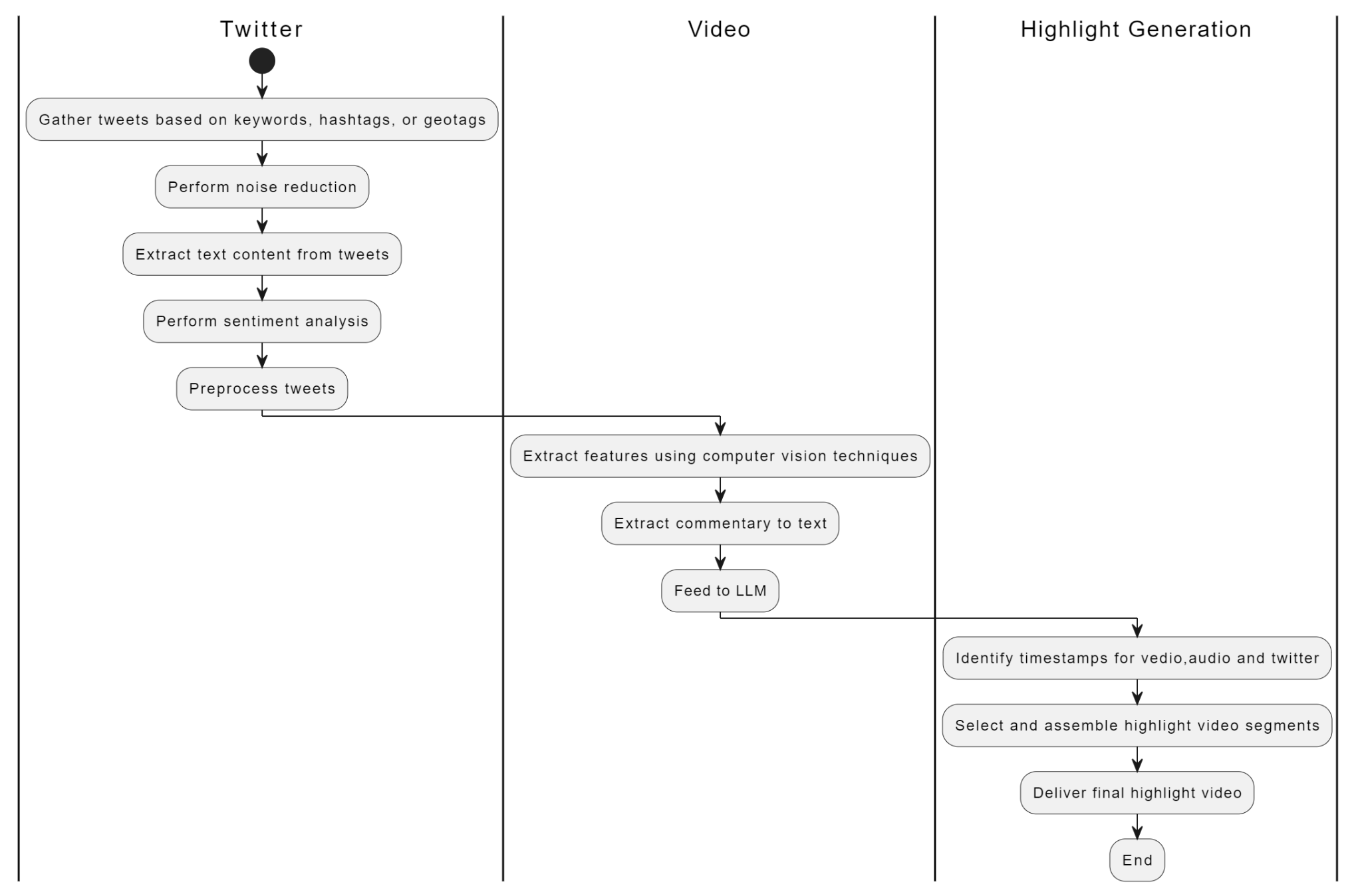
Development of reusable modules within the project that can be used across different functionalities:

Generic data processing utilities (e.g., data cleaning, timestamp handling)

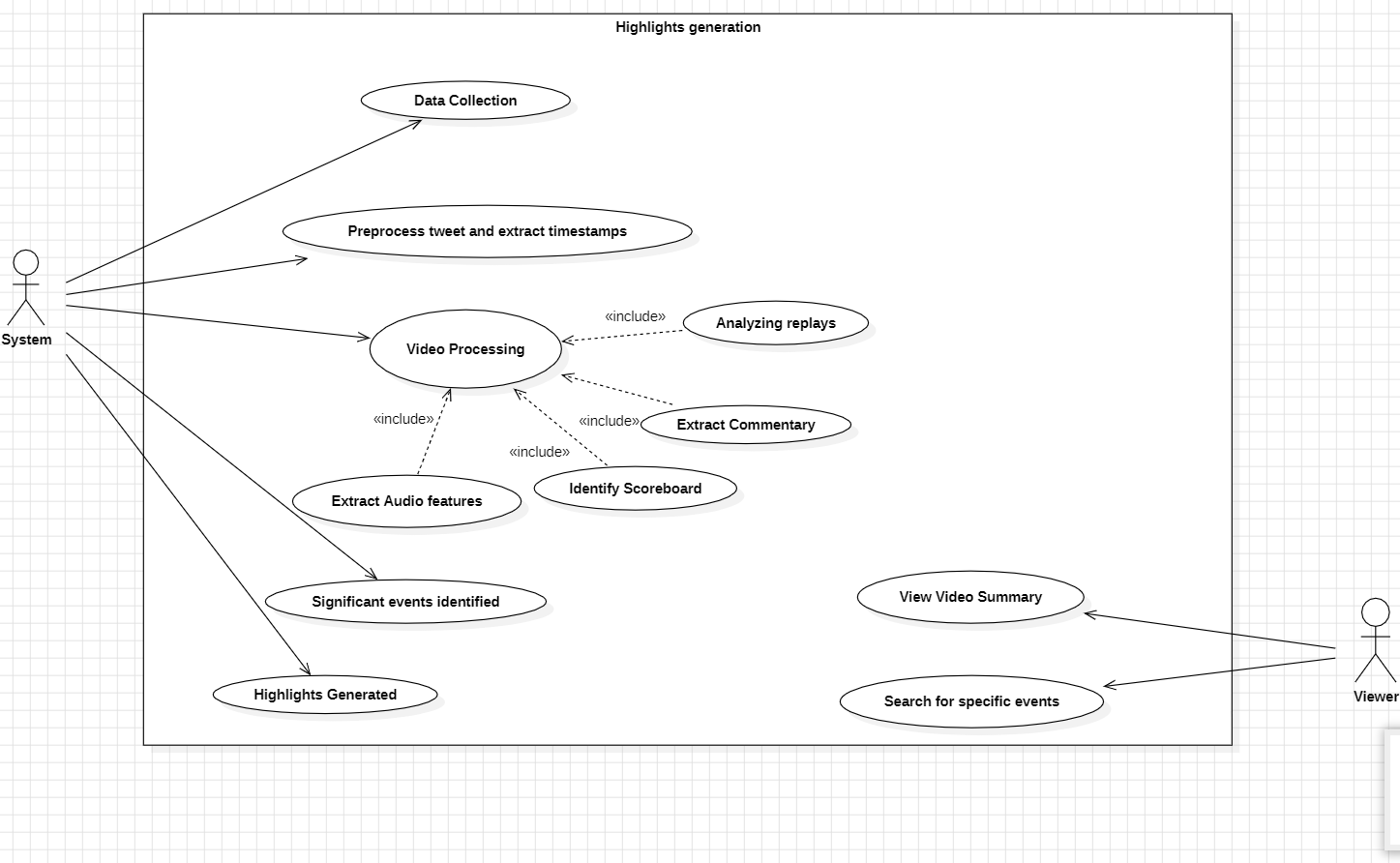
Event detection algorithms (e.g., WDHG approach implementation)

Highlight generation logic.

**ER Diagram / Swimlane Diagram / State Diagram (include as appropriate)**

****

**User Interface Diagrams**



**Actors:**

* **Viewer:** The viewer is the primary user of the system and the most likely only actor

explicitly shown in the diagram. They interact with the system to trigger various

functionalities.

* **System:**The system is the secondary use case of this model and it interacts with the

Twitter data and video and get the timestamps for the key events

**Use Cases:**

* **Data Collection:** This use case represents the process of gathering data for video analysis.
* Collecting tweets containing text or video related to the sport.
* Gathering the sports video itself from a source (live broadcast, pre-recorded footage).
* **Video Processing:** This use case signifies the analysis of the sports video. The diagram might include smaller elements within this use case that represent sub-processes:
* **Extract Commentary:** The system extracts audio commentary from the video to potentially understand the flow of the game and identify important moments based on the commentator's emphasis.
* **Identify Scoreboard:** The system identifies and extracts information from the scoreboard within the video, potentially capturing game stats or the current score.
* **Extract Audio Features:** This sub-process suggests the system analyzes audio features, perhaps crowd noise or music, to help identify exciting moments in the game (e.g., cheers for a goal)

.

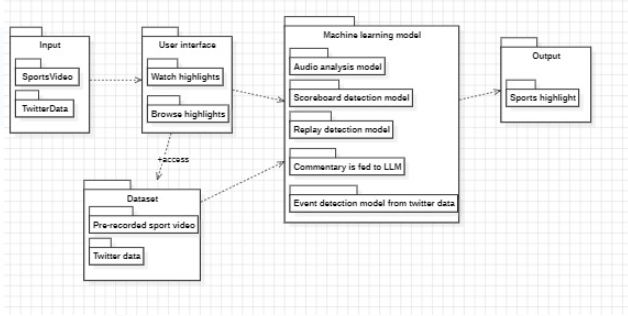
* **Analyzing Replays:** This use case refers to the system's ability to analyze specific replays within the video footage. Details might not be shown, but it could involve analyzing replays of key plays, controversial calls, or other significant moments.
* **View Video Summary:** This core functionality use case allows the viewer to interact with the system and view the generated highlight video containing the most important or exciting moments from the game.
* The viewer initiates data collection and video processing.
* The system analyzes the data and video using computer vision and potentially audio analysis techniques.
* Based on the analysis, the system generates a highlight video containing key moments.
* The viewer interacts with the system to view the generated highlight video.

**External Interfaces**

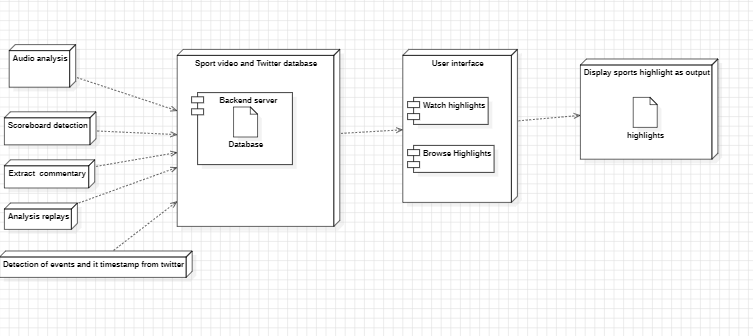
* The system shall provide a user-friendly graphical user interface (GUI) accessible via web or desktop application.
* Required screen formats shall adhere to GUI standards for styles, ensuring consistency and ease of use.
* Screen layouts shall be intuitive, with standard functions such as navigation menus, search bars, and help documentation available.
* Error messages shall be clear, concise, and descriptive, guiding users in resolving issues effectively.

**Packaging and Deployment Diagram**

**Packaging Diagram**

****

**Deployment Diagram**



* We take input as sports video and twitter data, then we detect the events .
* We use different machine learning models like audio analysis model, scoreboard detection model, replay detection model and commentary detection model. From these models we detect the events from sport videos.
* We also detect the events along with its timestamp from the twitter data .
* Merging the detected events from the twitter data and video input we create the summarization of sport videos.

# Help

To aid users in effectively utilizing the system, a comprehensive help system and documentation will be provided, including:

1. **User Manual**: A detailed user manual will be made available, providing comprehensive instructions on how to use the system's features and functionalities. The manual will cover topics such as account setup, video playback controls, search functionality, customization options, and accessing real-time insights from social media platforms like Twitter. It will include screenshots, diagrams, and examples to illustrate key concepts and procedures.
2. **FAQs and Troubleshooting Guide**: A collection of frequently asked questions (FAQs) and a troubleshooting guide will be compiled to address common queries and issues encountered by users. This resource will cover topics such as account management, data synchronization, playback errors, and troubleshooting tips for resolving technical issues.
3. **Video Tutorials**: To complement written documentation, a series of video tutorials will be created to demonstrate key features and workflows within the system. These tutorials will provide visual walkthroughs and practical examples to help users better understand how to navigate the system and perform various tasks effectively.
4. **Feedback Mechanism**: A feedback mechanism will be implemented to allow users to provide input, suggestions, and report issues with the system. Feedback channels such as online forms, email support, and community forums will be established to encourage user engagement and continuous improvement of the system based on user feedback.

By providing a range of help resources and documentation, users will have access to the information and assistance they need to maximize their utilization of the system, troubleshoot issues effectively, and enhance their overall experience with the platform.

# Appendix A: Definitions, Acronyms and Abbreviations

| 1. | GUI | Abbreviation for Graphical User Interface, which encompasses all aspects of a software application or system that users interact with, including screens, buttons, and menus. |
| --- | --- | --- |
| 2. | ML | Abbreviation for Machine Learning, a subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed. |
| 3. | API | Abbreviation for Application Programming Interface, which defines a set of rules and protocols for building and interacting with software applications. |
| 4. | CPU | Abbreviation for Central Processing Unit, the primary component of a computer that performs instructions and processes data. |
| 5. | GPU | Abbreviation for Graphics Processing Unit, a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images for display on a screen. |
| 6. | API Integration | The process of connecting different software systems or components to enable them to work together seamlessly and share data. |
| 7. | Database | A structured collection of data that is organized and stored in a computer system, typically for easy access and manipulation. |
| 8. | Security Measures | Techniques and protocols implemented to protect computer systems and data from unauthorized access, cyberattacks, and data breaches. |
| 9. | Real-Time | Refers to systems or processes that are capable of processing and delivering data or information instantaneously or with minimal delay. |
| 10. | User Authentication | The process of verifying the identity of a user to ensure that they are who they claim to be before granting access to a system or application. |
| 11. | Data Encryption | The process of converting data into a code to prevent unauthorized access or interception during transmission or storage. |
| 12. | User Feedback | Input or comments provided by users regarding their experiences, preferences, or suggestions for improvement related to a product or service. |
| 18. | API Documentation | A comprehensive reference guide that provides information on how to use and interact with an API, including endpoints, parameters, and authentication requirements. |

# Appendix B: References

[1] Hirasawa K, Maeda K, Ogawa T, Haseyama M. Detection of Important Scenes in Baseball Videos via a Time-Lag-Aware Multimodal Variational Autoencoder. Sensors (Basel). 2021 Mar 14;21(6):2045. doi: 10.3390/s21062045. PMID: 33799412; PMCID: PMC7999231.

[2] A. Javed, K. B. Bajwa, H. Malik and A. Irtaza, "An Efficient Framework for Automatic Highlights Generation from Sports Videos," in *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 954-958, July 2016, doi: 10.1109/LSP.2016.2573042

[3] H. Sattar, M. S. Umar, E. Ijaz and M. U. Arshad, "Multi-Modal Architecture for Cricket Highlights Generation: Using Computer Vision and Large Language Model," *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*, 2023, pp. 1-6, doi: 10.1109/ICOSST60641.2023.10414235

[4] Z. Saeed, R. Ayaz Abbasi, M. I. Razzak and G. Xu, "Event Detection in Twitter Stream Using Weighted Dynamic Heartbeat Graph Approach [Application Notes]," in *IEEE Computational Intelligence Magazine*, vol. 14, no. 3, pp. 29-38, Aug. 2019, doi: 10.1109/MCI.2019.2919395

# Appendix C: Record of Change History

[This section describes the details of changes that have resulted in the current High-Level Design document.]

| **#** | **Date** | **Document Version No.** | **Change Description** | **Reason for Change** |
| --- | --- | --- | --- | --- |
|  | 8/4/24 | 1 | Changes in the high level design diagram | Removed the redundant parts. |
|  | 10/4/24 | 2 | Changes in the description of use-case diagram | Added additional functionality |
|  | 12/4/24 | 3 | Changes in the deployment diagram | changed the functionality of 2 node |

# Appendix D: Traceability Matrix

| **Project Requirement Specification Reference Section No. and Name.** | **DESIGN / HLD Reference Section No. and Name.** |
| --- | --- |
| 3.2 User classes and Characteristics | 5.1 Master Class diagram |
| 4 Functional requirements | 4 High Level System Design |
| 5.2 Hardware Requirements | 9 Packaging and Deployment Diagram |
| 5.1 User interfaces | 7 User Interface Diagram |