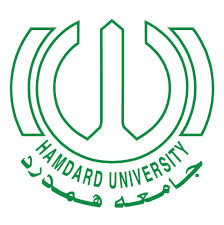
Hamdard University

Department of Computing

Final Year Project



**SpinMaster**

**FYP-035/FL24**

**Software Design Specifications**

Submitted by

**Syed Abraash Ul Haq**

BSAI/F21/2269

**Mubashir Naveed**

BSAI/F21/2433

Supervisor

**Mr. Amir Hussain**

Co Supervisor

**Mr. Waqas Pasha**

**Fall 2021**

**Document Sign off Sheet**  
**Document Information**

|  |  |
| --- | --- |
| **Project Title** | Spin Master |
| **Project Code** | FYP-035/FL24 |
| **Document Name** | Software Requirements Specifications |
| **Document Version** | 1.0 |
| **Document Identifier** | FYP-035/FL24-SDS |
| **Document Status** | Final |
| **Author(s)** | Abraash Syed Ul Haq & Mubashir Naveed |
| **Approver(s)** | Mr. Amir Hussain |
| **Issue Date** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Signature** | **Date** |
| Syed Abraash ul Haq | Team Lead |  |  |
| Mubashir Naveed | Team Member 2 |  |  |
| Muntaha Mehboob | Supervisor |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  | 1 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Definition of Terms, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **AI Model** | A computational model built using machine learning techniques to solve specific tasks like shot classification, movement analysis, and rule detection. |
| **AI** | **Artificial Intelligence**: A branch of computer science that focuses on creating machines capable of intelligent behavior, such as learning and problem-solving. |
| **API** | **Application Programming Interface**: A set of rules that allows different software applications to communicate with each other. |
| **AWS** | **Amazon Web Services**: A comprehensive cloud computing platform provided by Amazon, offering storage, computing power, and machine learning tools. |
| **CV** | **Computer Vision**: A field of computer science that enables machines to interpret and understand visual information from the world, such as images and videos. |
| **GPU** | **Graphics Processing Unit**: A hardware component specialized in performing complex calculations required for tasks like rendering graphics and processing data in parallel. |
| **IDE** | **Integrated Development Environment**: A software application that provides tools for writing, testing, and debugging code in one interface. |
| **ML** | **Machine Learning**: A subset of AI that involves algorithms and statistical models that allow systems to improve their performance on tasks through experience. |
| **RACI** | **Responsible, Accountable, Consulted, and Informed**: A matrix that defines the roles and responsibilities of team members for project tasks. |
| **SQL** | **Structured Query Language**: A programming language used for managing and querying data in relational databases. |
| **UI** | **User Interface**: The interface through which users interact with the system, consisting of the visual elements and controls. |
| **Verification and Validation (V&V)** | A methodology used to ensure that a system meets its requirements and performs its intended functions effectively. |

# Table of Contents

[1.1.1 Document Information 2](#_Toc187605890)

[2 Table of Contents 4](#_Toc187605891)

[3 Introduction 4](#_Toc187605892)

[3.1 Purpose of Document 5](#_Toc187605893)

[3.2 Intended Audience 5](#_Toc187605894)

[3.3 Document Convention 5](#_Toc187605895)

[3.4 Project Overview 5](#_Toc187605896)

[3.5 Scope 5](#_Toc187605897)

[4 Design Considerations 7](#_Toc187605898)

[4.1 Assumptions and Dependencies 7](#_Toc187605899)

[4.2 Risks and Volatile Areas 7](#_Toc187605900)

[4.3 System Level Architecture 8](#_Toc187605901)

[4.4 Software Architecture 8](#_Toc187605902)

[5 Design Strategy 9](#_Toc187605903)

[6 Detailed System Design 11](#_Toc187605904)

[6.1 Design Class Diagram 11](#_Toc187605905)

[6.2 Database Design 12](#_Toc187605906)

[6.2.1 ER Diagram 12](#_Toc187605907)

[6.2.2 Data Dictionary 13](#_Toc187605908)

[6.3 Application Design 20](#_Toc187605909)

[6.3.1 Sequence Diagram 20](#_Toc187605910)

[6.3.2 State Diagram 21](#_Toc187605911)

[6.4 GUI Design 21](#_Toc187605912)

[6.4.1 <Use Case Name - Mock Screen 1> 21](#_Toc187605913)

[6.4.2 <Use Case Name - Mock Screen 2> 21](#_Toc187605914)

[6.4.3 <Use Case Name - Mock Screen 3> 21](#_Toc187605915)

[7 References 22](#_Toc187605916)

[8 Appendices 23](#_Toc187605917)

### 1 Introduction

The Advanced Table Tennis Analytics System leverages the power of computer vision and machine learning to analyze recorded match footage and provide detailed insights into player performance and match events. This project focuses on developing a non-real-time analytics platform that enables coaches, players, and enthusiasts to understand the dynamics of table tennis games through in-depth metrics and visualizations. By processing recorded video data, the system can detect key events, classify shots, and generate comprehensive performance reports. This innovative solution bridges the gap between raw gameplay footage and actionable insights, fostering improved strategies and decision-making in the sport.

#### 1.1 Purpose of the Document

This Software Design Specification (SDS) outlines the design, architecture, and functionalities of the Advanced Table Tennis Analytics System. The project focuses on extracting insights from recorded table tennis matches using computer vision and machine learning techniques. The system analyzes player performance, identifies key match events, and provides data visualization for post-match review.

#### 1.2 Intended Audience

The intended audience for this document includes developers and data scientists involved in building the system, as well as coaches and players seeking to enhance performance through analytics. Academic researchers in the domain of sports analytics and project stakeholders will also benefit from the comprehensive information provided in this document.

**1.3 Document Convention**

This document uses:

* **Font:** Arial, size 12 for text, size 14 for headings.
* **Notations:** UML diagrams for system design, ER diagrams for database design.

#### 1.4 Project Overview

The system employs advanced technologies to analyze recorded video footage of table tennis matches. Key features include tracking player movements, classifying shot types, detecting match events, and visualizing performance metrics. Unlike real-time systems, this project focuses on generating detailed reports post-match, offering in-depth insights and actionable recommendations.

#### 1.5 Scope

##### 1.5.1 In-Scope

* Player tracking using computer vision techniques.
* Shot classification (e.g., forehand, backhand).
* Match event detection.
* Performance metrics visualization.
* Report generation for post-match review.

##### 1.5.2 Out-of-Scope

* Real-time analysis of match footage.
* Integration with live broadcasting systems.
* Predictive analytics for future matches.

**2. Design Considerations**

Spin master is designed to offer a user-friendly experience with simple navigation and a clean, minimalist interface for easy video uploads and match analysis. The platform will be responsive, ensuring a consistent experience across devices. Accessibility features include screen reader support, keyboard navigation, and high-contrast text for better readability.

The system will optimize the processing of pre-recorded videos using cloud-based infrastructure and machine learning models to analyze data efficiently. It will comply with privacy regulations, encrypt sensitive data, and implement role-based access for security.

Scalability is ensured with cloud services and independent scaling of features like video analysis. User feedback will drive continuous improvements, with updates based on data and community engagement to refine the platform.

#### 2.1 Assumptions and Dependencies

Assumptions are critical to the development and functionality of the Advanced Table Tennis Analytics System. It is assumed that high-quality video footage will be available, ensuring consistent frame rates and resolution. Proper calibration of the court and camera angles is expected to enhance the system’s accuracy. Additionally, reliable hardware and cloud storage solutions will facilitate computationally intensive tasks and data management. Compliance with relevant data privacy regulations will safeguard user data and ensure ethical handling of sensitive information.

##### 2.1.1 Video Quality and Consistency

It is assumed that high-quality video footage with a consistent frame rate is available for analysis. Variations in resolution or frame rates could adversely affect the accuracy of player and ball detection. Ensuring consistent video quality is critical to maintaining reliable analytics results.

##### 2.1.2 Court and Camera Calibration

It is presumed that the system will initially be calibrated manually to account for table dimensions, camera angles, and player positions. This calibration step ensures accurate spatial measurements and event detection throughout the match analysis.

##### 2.1.3 Hardware Requirements

The hardware utilized for data processing must meet minimum specifications to handle computationally intensive tasks like video processing and machine learning inference. GPUs or high-performance computing infrastructure may be required for large-scale analyses.

##### 2.1.4 Machine Learning Model Training

The system depends on the availability of a labeled dataset of table tennis shots and events for training and validation of machine learning models. The quality and diversity of this dataset will significantly impact the accuracy of the system.

##### 2.1.5 Cloud Storage and Processing

The project relies on cloud services such as AWS S3 for storing large video files and processed analytics. Reliable and scalable cloud infrastructure is essential for efficient storage and retrieval.

##### 2.1.6 Internet Connectivity

Stable internet connectivity is assumed to facilitate the upload of match footage and retrieval of analytics data. Interruptions in connectivity could delay processing and user access to results.

#### 2.2 Risks and Challenges

##### 2.2.1 Motion Blur and Object Occlusion

Fast-moving objects, such as the ball, can appear blurred in footage, making detection and tracking difficult. Additionally, the ball may become temporarily occluded by players or rackets, requiring robust prediction algorithms to handle such scenarios.

##### 2.2.2 Variability in Lighting Conditions

Differences in lighting, shadows, and reflections in video footage can interfere with computer vision algorithms. The system must include preprocessing steps like histogram equalization and adaptive thresholding to mitigate these effects.

##### 2.2.3 Player Behavior and Shot Diversity

Unconventional player movements or shot techniques may reduce the accuracy of machine learning models. Regular updates and retraining of the models with new data are necessary to improve adaptability to diverse gameplay styles.

##### 2.2.4 Data Privacy and Security Risks

Storing and analyzing sensitive match footage and player data poses privacy concerns. Ensuring encrypted storage and secure access controls is critical to protecting data integrity and maintaining user trust.

##### 2.2.5 System Scalability

As the volume of matches and analytics requests increases, the system must scale to handle additional workloads. Efficient use of resources and implementation of load balancing will be essential for scalability.

**2.2.6 Delayed Feedback Loop**

The non-real-time nature of the system means there is a delay between footage upload and result generation. Users must be informed of expected processing times, and system efficiency must be optimized to minimize delays.

##### 2.2.7 Model Interpretability

Providing clear and interpretable analytics results is a challenge, particularly for non-technical users. Visualizations and simplified summaries must effectively communicate complex insights.

**3. System Architecture**

The system architecture for Spin masteris designed to provide an efficient analytics solution for table tennis match analysis, consisting of several layers to ensure smooth operation:

**1. Client Layer (Frontend)**

• **Components**: Web interface.

• **Technology**: HTML / CSS.

• **Responsibilities**:

* Provides an interactive dashboard for displaying match stats and analytics.
* Allows users to upload videos and view real-time results.
* Communicates with the backend via APIs.

**2. Application Layer (Backend)**

• **Components**: Server-side app, and ML models.

• **Technology**: Python, Flask/Django.

• **Responsibilities**:

* Handles data processing, including video analysis and player tracking.
* Runs machine learning models for performance predictions.
* Provides bridge for data retrieval.

**3. Database Layer**

• **Components**: Relational/NoSQL database.

• **Responsibilities**:

* Stores match data, player stats, and analysis results.
* Logs user interactions and analysis reports.

**4. Computer Vision & Machine Learning Layer**

• **Components**: OpenCV, TensorFlow.

• **Responsibilities**:

* Analyzes videos for player movement, ball tracking, and shot classification.
* Uses ML for performance predictions and shot outcome analysis.

**5. Data Processing & Video Analysis Layer**

• **Components**: Video processing tools.

• **Responsibilities**:

* Processes match videos to extract frames and track movements.
* Outputs data for further analysis.

**6. Reporting & Analytics Layer**

• **Components**: Python 3.10+.

• **Responsibilities**:

* Provides analytics and reports.
* Supports exporting reports in various formats.

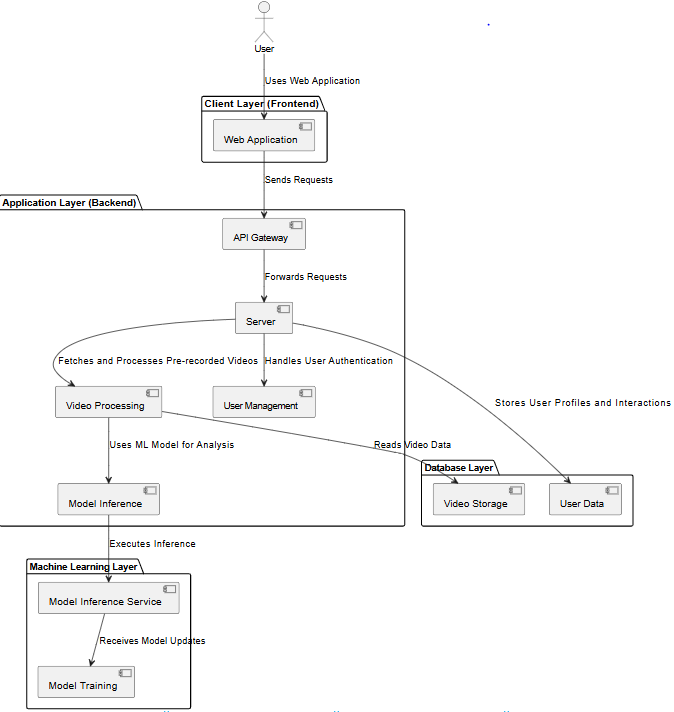
**7. Security Layer**

• **Components**: Authentication tools (OAuth, JWT).

• **Responsibilities**:

* Secures user access and data with encryption and role-based access control.

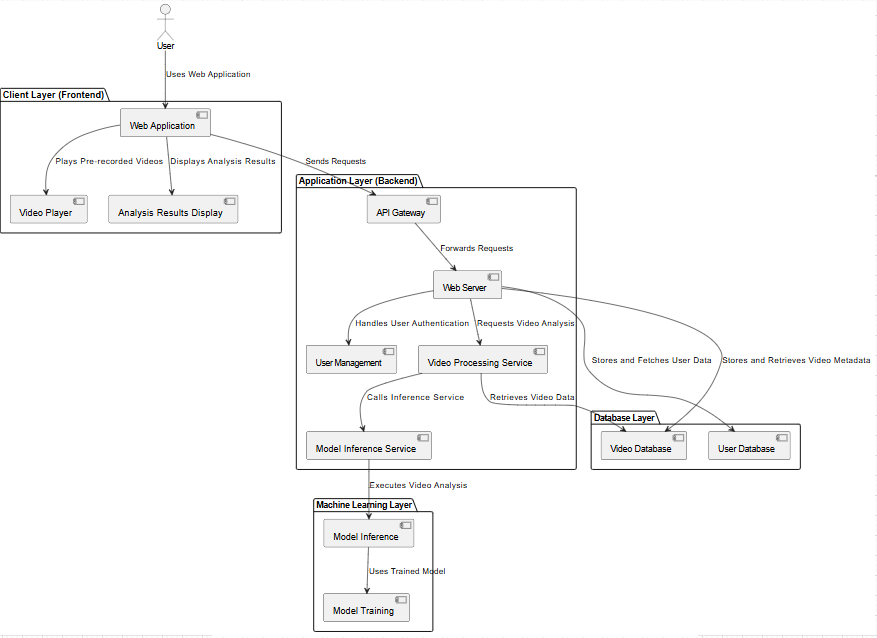
## 3.1 System Level Architecture



**Figure 1: System Level Architecture**

1. **Client Layer (Frontend)**: The interface through which users interact with the system. In this case, the web application.
2. **Application Layer (Backend)**: The core logic and service layer that processes user requests.
3. **Database Layer**: Stores all data related to users and videos.
4. **Machine Learning Layer**: The layer that handles model training and inference.
5. **Video Storage**: Stores the video files and metadata.

## 3.2 Software Architecture



* **Figure 2: System Level Architecture**
* **Client Layer (Frontend)**: The web interface for user interaction.
* **Application Layer (Backend)**: Handles business logic and connects frontend with backend services.
* **Database Layer**: Stores user and video data.
* **Machine Learning Layer**: Analyzes videos using trained models.

# 4 Design Strategy

* **Design Strategy**

The system is designed for flexibility, scalability, and efficiency, focusing on providing a high-quality user experience and computational performance while working with pre-recorded gameplay videos.

* **Future Extensions**

The modular system allows for future enhancements like incorporating more advanced machine learning models, better analysis techniques, or adding features like real-time match analysis when needed in the future.

* **System Reuse**

Key components such as video preprocessing, ball tracking, and machine learning models are designed for reuse across different components, ensuring that foundational work can be applied to other aspects of the project or future projects.

* **User Interface Design**

The UI is intuitive, user-centric, and adaptable to varying levels of expertise. It displays video analysis results clearly with actionable insights, offering both in-depth analytics for advanced users and simplified visualizations for beginners.

* **Data Management**

Data is stored in the cloud to support scalability and quick access to video data and analysis results. The system uses efficient data management techniques like compression and caching to reduce storage and improve performance.

* **Concurrency & Synchronization**

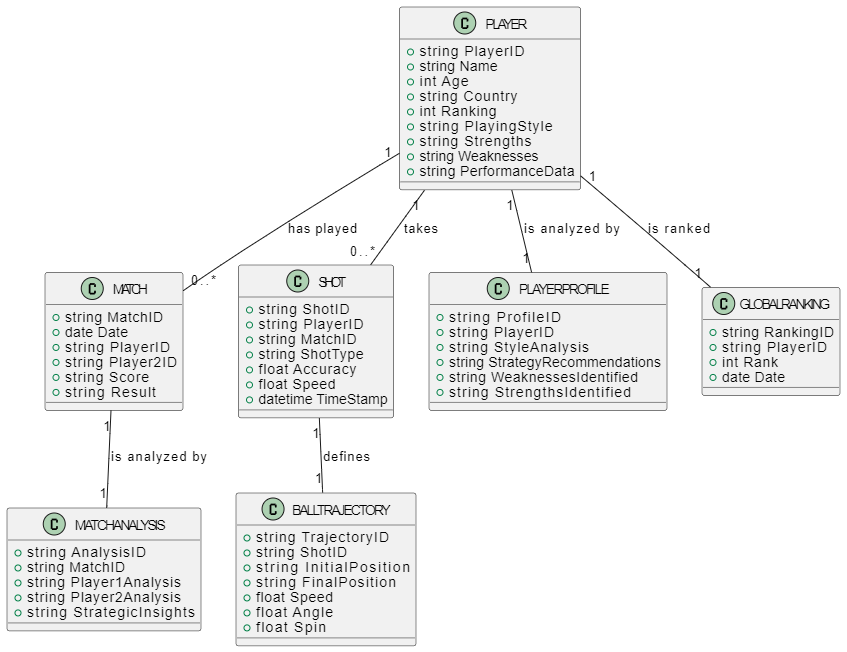
The system processes tasks concurrently, ensuring efficient handling of multiple videos and analysis tasks, with synchronization techniques in place to ensure consistency when managing data.

* **Design Trade-offs**

The design balances computational complexity with the need for accurate post-processed analysis of pre-recorded gameplay videos, offering scalable insights while ensuring the system remains user-friendly for both beginners and advanced users.

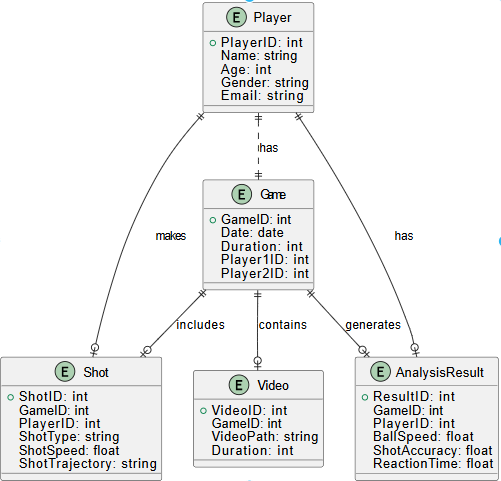
# Detailed System Design

## 4.1 Design Class Diagram



**Figure 3: Design Class Diagram**

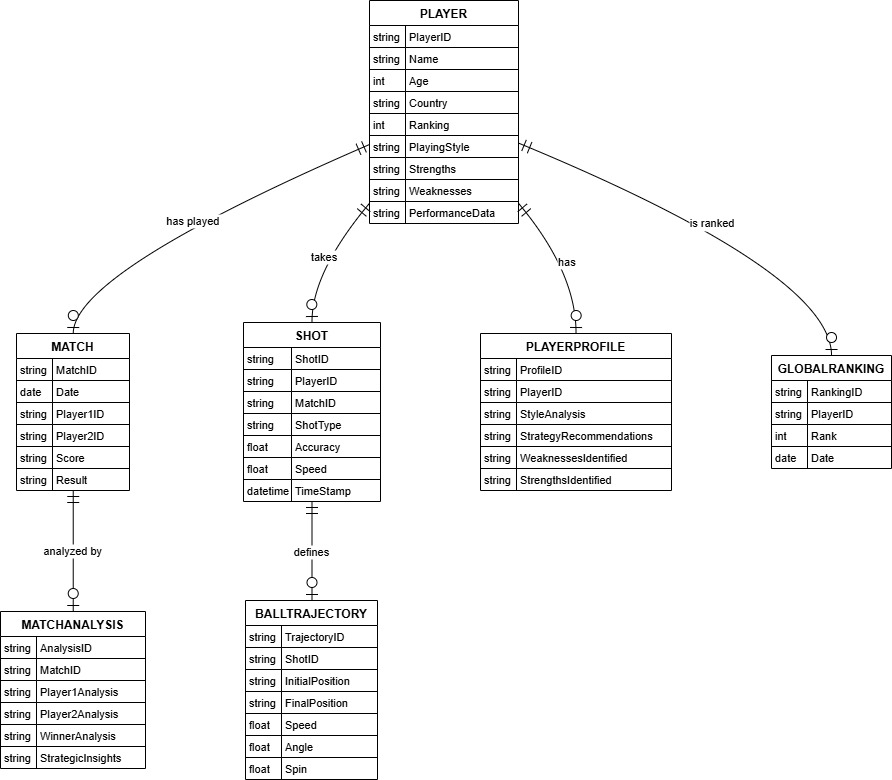
## 4.2 Database Design



**Figure 4: Database Design**

* **Player**: Stores player info (ID, Name, Age, Gender, Email).
* **Game**: Represents a match between two players (ID, Date, Duration, Player1/Player2).
* **Shot**: Tracks shot details (ID, Type, Speed, Trajectory).
* **Analysis Result**: Stores performance results (Ball Speed, Accuracy, Reaction Time).

### ER Diagram



**Figure 5: ER Diagram**

* **Player**: Stores player info (PlayerID, Name, Age, Gender, Email).
* **Match**: Represents a match (MatchID, Date, Duration, Player1ID, Player2ID).
* **Player Profile**: Stores detailed profile data (ProfileID, PlayerID, Bio, Stats).
* **Shot**: Represents shots in a match (ShotID, MatchID, PlayerID, Type, Speed, Trajectory).
* **Global Ranking**: Stores player's global ranking (PlayerID, Rank, Points).
* **Match Analysis**: Stores analysis data of the match (AnalysisID, MatchID, PlayerID, BallSpeed, Accuracy).
* **Ball Trajectory**: Represents ball movement data (TrajectoryID, ShotID, Path, Speed).

### Data Dictionary

|  |  |
| --- | --- |
| Match | |
| **Name** | MATCH. |
| **Alias** | matches. |
| **Where-used/how-used** | Used in PLAYER, SHOT, MATCH\_ANALYSIS |
| **Content description** | Contains information about all players in the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Match\_ID | Unique identifier for each match | String | 10 | No | (Auto-generated) | Primary Key |
| Date | Date of the match | Date | 8 | No |  |  |
| Player1ID | ID of the first player | String | 10 | No |  | FK (references Player\_ID in PLAYER) |
| Player2ID | ID of the second player | String | 10 | No |  | FK (references Player\_ID in PLAYER) |
| Score | Match score (e.g., 3-1) | String | 10 | No | 0 |  |
| Result | Outcome of the match (e.g., Win, Loss) | String | 10 | No |  |  |

|  |  |
| --- | --- |
| Shot | |
| **Name** | SHOT. |
| **Alias** | shots. |
| **Where-used/how-used** | Used in PLAYER, BALL\_TRAJECTORY |
| **Content description** | Contains information about each shot played during a match. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Shot\_ID | Unique identifier for each shot | String | 10 | No | (Auto-generated) | Primary Key |
| Player\_ID | ID of the player who took the shot | Date | 10 | No |  | FK (references Player\_ID in PLAYER) |
| Match\_ID | ID of the match the shot was played in | String | 10 | No |  | FK (references Match\_ID in MATCH) |
| Shot type | Type of shot (e.g., Forehand, Backhand, Serve) | String | 20 | No |  |  |
| Accuracy | Accuracy of the shot (e.g., percentage) | Float | 5,2 | Yes | 0 |  |
| Speed | Speed of the shot | Float | 5,2 | Yes | 0 |  |
| Time stamp | Timestamp of the shot | Timestamp | 23 | No |  |  |

|  |  |
| --- | --- |
| PLAYER\_PROFILE | |
| **Name** | PLAYER\_PROFILE |
| **Alias** | Player profiles. |
| **Where-used/how-used** | Used in PLAYER |
| **Content description** | Contains detailed information about a player's profile. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Profile\_ID | Unique identifier for each player profile | String | 10 | No | (Auto-generated) | Primary Key |
| Player\_ID | ID of the player associated with the profile | String | 10 | No |  | FK (references Player\_ID in PLAYER) |
| Style Analysis | Analysis of the player's playing style | Text | (variable) | Yes |  |  |
| Strategy  Recommendations | Recommended strategies for the player | Text | (variable) | Yes |  |  |
| Weaknesses  Identified | Identified weaknesses of the player | Text | (variable) | Yes | 0 |  |
| Strengths  Identified | Identified strengths of the player | Text | (variable) | Yes | 0 |  |

|  |  |
| --- | --- |
| GLOBAL\_RANKING | |
| **Name** | GLOBAL\_RANKING |
| **Alias** | Global Rankings. |
| **Where-used/how-used** | Used in PLAYER. |
| **Content description** | Contains information about a player's global ranking. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Ranking\_ID | Unique identifier for each player profile | String | 10 | No | (Auto-generated) | Primary Key |
| Player\_ID | ID of the player associated with the ranking | String | 10 | No |  | FK (references Player\_ID in PLAYER) |
| Rank | Current global ranking of the player | Integer | 5 | No |  |  |
| Date | Date of the ranking | Date | 8 | No |  |  |

|  |  |
| --- | --- |
| MATCH\_ANALYSIS | |
| **Name** | MATCH\_ANALYSIS |
| **Alias** | Match Analyses |
| **Where-used/how-used** | Used in MATCH |
| **Content description** | Contains analysis of a match. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Analysis\_ID | Unique identifier for each match analysis | String | 10 | No | (Auto-generated) | Primary Key |
| Match\_ID | ID of the match analyzed | String | 10 | No |  | FK (references Match\_ID in Match) |
| Player1  Analysis | Analysis of the first player's performance | Text | (Variable) | Yes |  |  |
| Player2  Analysis | Analysis of the second player's performance | Text | Variable) | Yes |  |  |
| Winner Analysis | Analysis of the winning player's performance | Text | (Variable) | Yes |  |  |
| Strategic Insights | Strategic insights from the match | Text | (Variable) | Yes |  |  |

|  |  |
| --- | --- |
| BALL\_TRAJECTORY | |
| **Name** | BALL\_TRAJECTORY |
| **Alias** | Ball Trajectories |
| **Where-used/how-used** | Used in SHOT |
| **Content description** | Contains information about the ball trajectory of a shot. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Trajectory\_ID | Unique identifier for each ball trajectory | String | 10 | No | (Auto-generated) | Primary Key |
| Shot\_ID | ID of the shot associated with the trajectory | String | 10 | No |  | FK (references Shot\_ID in Shot) |
| Initial  Position | Initial position of the ball | String | 50 | No |  |  |
| Final Position | Final position of the ball | String | 50 | No |  |  |
| Speed | Speed of the ball | Float | 5,2 | Yes | 0 |  |
| Angle | Angle of the ball's trajectory | Float | 5,2 | Yes | 0 |  |
| Spin | Spin of the ball | Float | 5,2 | Yes | 0 |  |

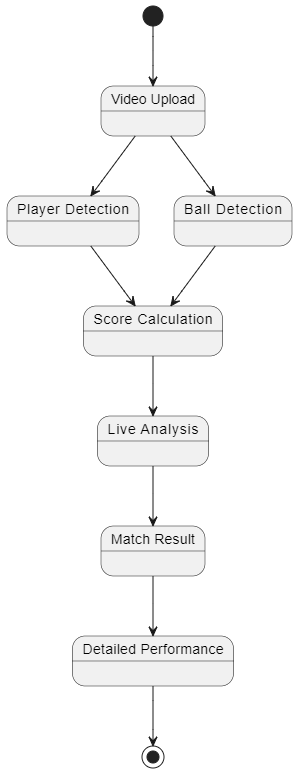
## Application Design

### Sequence Diagram



**Figure 6: Sequence Diagram**

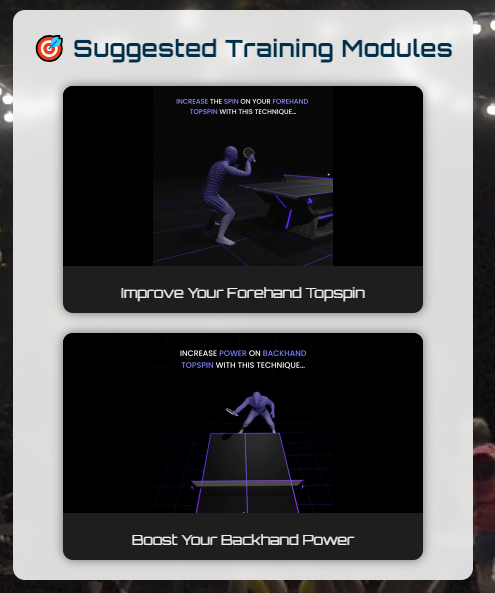
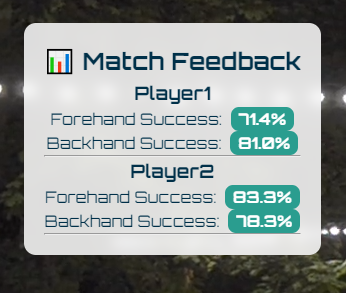
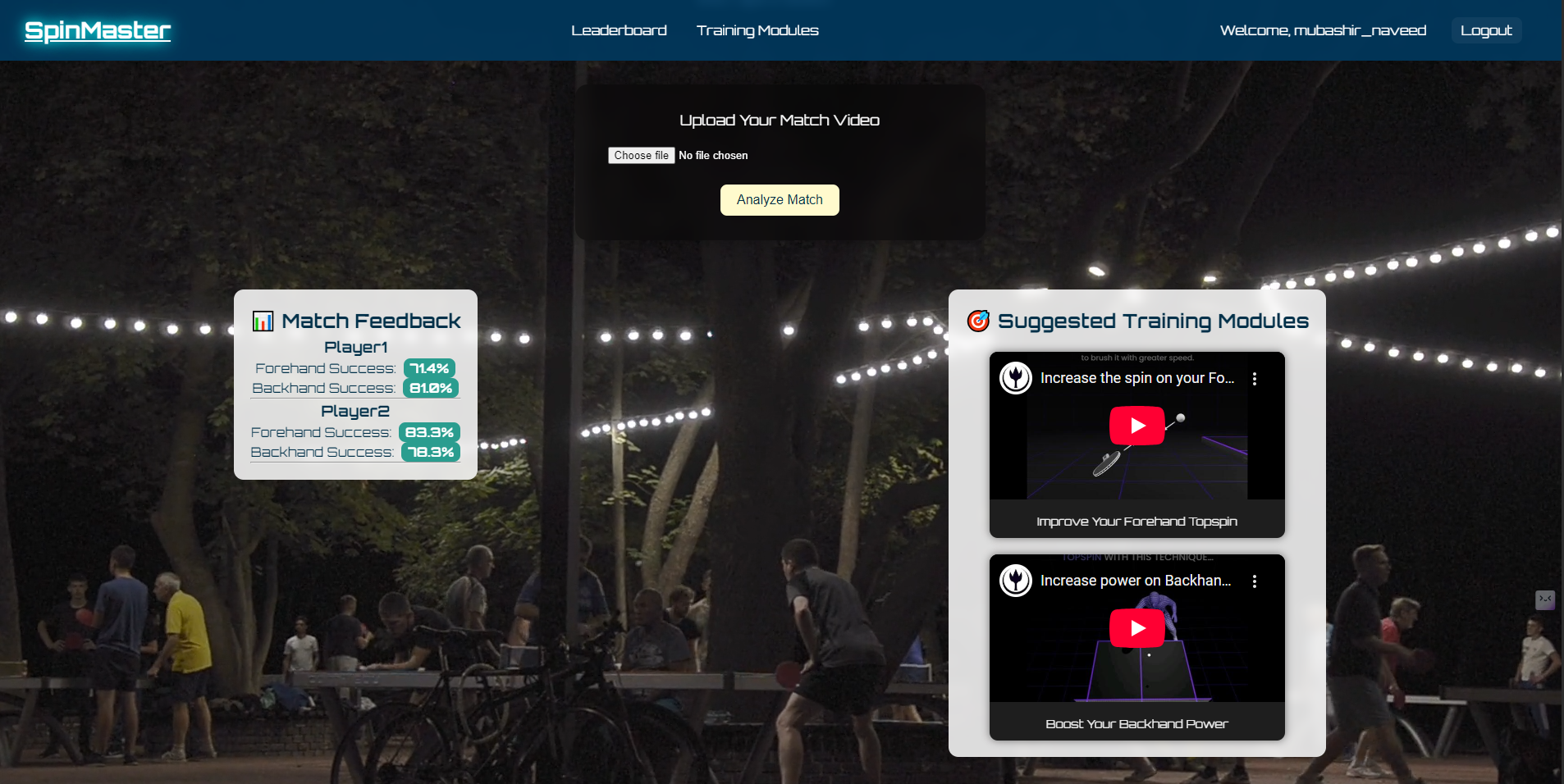
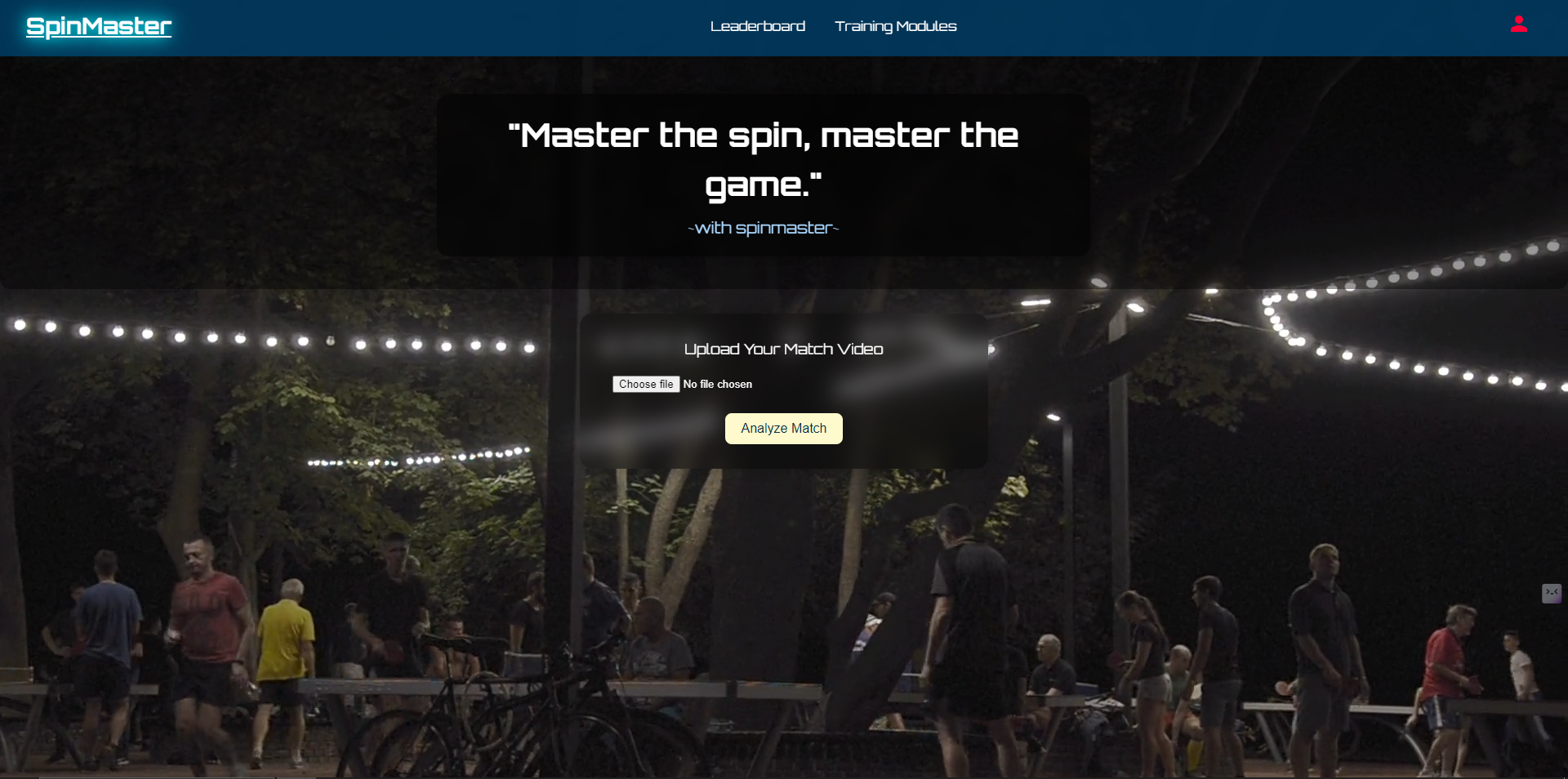
### State Diagram



**Figure 7: Sequence Diagram**

1. **Video Upload**: Upload the recorded match video.
2. **Player Detection**: Detect and track players in the video.
3. **Ball Detection**: Track the ball's movement in the video.
4. **Score Calculation**: Calculate points based on the game’s rules.
5. **Live Analysis**: Provide real-time analysis of the match (using pre-recorded data).
6. **Match Result**: Generate the final match outcome and scores.
7. **Detailed Performance**: Analyze player performance (accuracy, speed, etc.).

## 4.3 GUI Design



# References

1. **TTNet: Real-time Temporal and Spatial Video Analysis of Table Tennis**
   * Authors: Roman Voeikov, Nikolay Falaleev & Ruslan Baikulov
   * Published: April 2020
   * Relevance: This paper provides insights into designing real-time video analysis systems for table tennis, which aligns with SpinMaster's goal to analyze pre-recorded matches using temporal and spatial analysis of player movements and ball trajectories.
   * Source: [arXiv](https://arxiv.org/abs/2004.09927)
2. **A Multi-Modal Table Tennis Robot System**
   * Authors: Andreas Ziegler, Thomas Gossard, Karl Vetter, Jonas Tebbe & Andreas Zell
   * Published: October 2023
   * Relevance: This paper discusses vision-based detection and trajectory estimation in table tennis, which supports SpinMaster's focus on accurate tracking of ball spin and player movements using multiple cameras.
   * Source: [arXiv](https://arxiv.org/abs/2310.19062)
3. **Designing an Artificial Intelligence-based Sport Management System**
   * Authors: Junwei Feng
   * Published: 2023
   * Relevance: This paper's proposed AI-based management system integrates AI algorithms for performance analysis, which directly supports the SpinMaster project's design and architecture for structured analysis and feedback.
   * Source: [Springer Link](https://link.springer.com/article/10.1007/s00500-023-09162-0)
4. **Fast Moving Table Tennis Ball Tracking Algorithm Based on Graph Model and Trajectory Prediction**
   * Authors: Tianjian Zou, Jiangning Wei, Bo Yu, Xinzhu Qiu, Hao Zhang, Xu Du & Jun Liu
   * Published: 2024
   * Relevance: The graph-based ball tracking algorithm in this paper is highly relevant to SpinMaster's objective of implementing a robust ball trajectory and speed analysis system for table tennis.
   * Source: [Nature](https://www.nature.com/articles/s41598-024-80056-3)