

# NBA Data Analysis Project

## Requirements Document

*Tunahan Oğuz*  
*Ali Eren Kurt*  
*Alkım Doryan*  
*Beyzanur Zeybek*

### 1. Functional Requirements

The following are the detailed functionalities that the NBA Data Analysis System is expected to perform. Each requirement is defined in clear, measurable, and testable terms.

#### 1. Data Ingestion:

- *Requirement:* The system **must** be able to import data from multiple CSV or database files representing the following tables: `common_player_info`, `draft_combine_stats`, `draft_history`, `game`, `game_info`, `game_summary`, `inactive_players`, `line_score`, `officials`, `other_stats`, `play_by_play`, `player`, `team`, `team_details`, `team_history`, and `team_info_common`.
- *Testability:* Verify that each table is successfully loaded into the system without errors.

#### 2. Data Cleaning and Preprocessing:

- *Requirement:* The system **must** perform data cleaning by handling missing values, standardizing data formats, and correcting data inconsistencies across all tables.
- *Testability:* Validate that after preprocessing, no fields contain irregular formats or unexpected null values beyond acceptable thresholds.

#### 3. Data Integration:

- *Requirement:* The system **must** integrate the multiple tables into a unified data repository, ensuring proper relationships (e.g., linking player IDs across tables).
- *Testability:* Run sample join queries to ensure the integrity of linked data across tables.

#### 4. Exploratory Data Analysis (EDA):

- *Requirement:* The system **must** generate exploratory data analysis reports including descriptive statistics, correlation matrices, and initial visualizations (e.g., histograms, scatter plots) for key performance indicators.
- *Testability:* Review the EDA report outputs and verify that they include the required statistical summaries and visualizations.

#### 5. Visualization Dashboard:

- *Requirement:* The system **must** provide an interactive dashboard that allows users to visualize key metrics (e.g., player performance, game outcomes, team statistics) using dynamic charts and filters.
- *Testability:* Confirm that users can interact with the dashboard (filter, zoom, select different metrics) and that visualizations update in real time.

#### 6. **Predictive Modeling:**

- *Requirement:* The system **must** support the development and execution of predictive models (e.g., predicting game outcomes, player performance trends) using historical data, with functionalities to train, validate, and test models.
- *Testability:* Validate the model outputs by comparing predicted results with a designated test dataset and verifying performance metrics (accuracy, precision, recall, etc.).

#### 7. **Automated Reporting:**

- *Requirement:* The system **must** generate automated reports summarizing the findings from both the EDA and predictive modeling phases, and export these reports in common formats (e.g., PDF, HTML).
- *Testability:* Verify that reports are correctly generated and exported with complete and accurate content.

#### 8. **Data Export:**

- *Requirement:* The system **must** allow users to export the cleaned and processed data subsets to external files (e.g., CSV, Excel) for further analysis.
- *Testability:* Confirm that data export functionality generates correctly formatted files containing the expected data.

#### 9. **Query Functionality:**

- *Requirement:* The system **must** enable users to execute custom queries on the unified data repository via a user interface, allowing filtering and aggregation based on specific criteria (e.g., date ranges, teams, players).
- *Testability:* Execute sample queries and validate that the returned results match the query conditions.

#### 10. **User Access and Role Management:**

- *Requirement:* The system **must** support user authentication and role-based access control (if deployed as a web application), ensuring that only authorized users can access sensitive data and specific functionalities.
- *Testability:* Verify that user login works correctly and that role-based permissions restrict or grant access as defined.

## 2. **Non-Functional Requirements**

The following quality characteristics define how the NBA Data Analysis System should perform:

#### 1. **Performance:**

- *Requirement:* The system **must** respond to user queries or dashboard interactions within 2 seconds under normal operating conditions.
- *Testability:* Conduct performance tests to ensure that response times are within the specified threshold.

#### 2. **Scalability:**

- *Requirement:* The system **must** be scalable to accommodate an increasing volume of data without significant degradation in performance.
- *Testability:* Simulate an increased dataset load and monitor system performance and responsiveness.

### 3. Usability:

- *Requirement:* The system **must** feature a user-friendly interface that allows both technical and non-technical users to navigate, query data, and interpret visualizations easily.
- *Testability:* Conduct usability testing sessions with a representative group of users and gather feedback on interface ease-of-use.

### 4. Compatibility:

- *Requirement:* The system **must** be compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) and function consistently across them.
- *Testability:* Test the system across different browsers and operating systems to confirm consistent behavior and layout.

### 5. Security:

- *Requirement:* The system **must** ensure data security by implementing appropriate authentication, authorization, and encryption measures to protect sensitive information.
- *Testability:* Perform security audits and penetration tests to verify that unauthorized access is prevented and data is encrypted where necessary.

## 3. Task Assignments:

Detailed task assignments and phases can be found in the Gantt chart below.

 **Gantt chart**

### Rationale for Task Assignment

The task assignments were made by carefully considering the skills, interests, and learning objectives of each team member. This approach ensures a fair and effective distribution of responsibilities, maximizing productivity while providing opportunities for skill development.

- **Alignment with Expertise:** Each team member was assigned tasks that align with their existing strengths. For instance, Tunahan, who has experience with project coordination, was given the Kick-off Meeting task to ensure smooth initial discussions. Similarly, Beyzanur, who excels in documentation, was responsible for drafting the Requirements Documentation.
- **Balanced Workload:** The tasks were distributed evenly to avoid overburdening any single team member. Complex tasks like Review & Iteration of Requirements were assigned to multiple members, such as Allkim, to ensure collaborative input and efficiency.
- **Encouraging Growth:** Assignments also factored in learning opportunities. For example, a team member interested in improving their documentation skills was given an opportunity to work on Requirements Drafting. This allows for professional development while ensuring high-quality outputs.
- **Task Interdependencies:** Assignments were structured to facilitate seamless collaboration. Tasks that require iterative improvements, such as Requirement Reviews, were assigned to those who had prior involvement in documentation to maintain continuity.

By adopting this strategic approach, we ensured that all team members contribute effectively while also enhancing their skills. This distribution fosters teamwork, maintains efficiency, and promotes a productive workflow.

Team Members and Roles

The following table outlines the main roles of each team member in the project:

Team Member	Role
Tunahan Oğuz	Backend Developer/ Tester
Beyzanur Zeybek	Requirements Analyst/ Frontend Developer
Allkim Doryan	Project Manager/ Tester/ Scrum Master
Ali Eren Kurt	Frontend Developer/ Product Owner

These roles were assigned based on each member’s expertise and interests to ensure a smooth and efficient workflow throughout the project.

4. Document-Specific Task Matrix for Requirements Document

Task ID	Task Description	Team Member(s) Responsible	Completion Status	Notes
T1	Gathering functional requirements	Beyzanur Zeybek, Ali Eren Kurt	Completed	Functional requirements listed and structured
T2	Defining data ingestion requirements	Tunahan Oğuz, Alkim Doryan	Completed	Data sources and integration points identified
T3	Specifying data cleaning & preprocessing	Beyzanur Zeybek, Tunahan Oğuz	Completed	Handling missing values and standardization defined

T4	Defining data integration processes	Alkım Doryan, Ali Eren Kurt	Completed	Database relationships and linking rules documented
T5	Outlining exploratory data analysis (EDA) requirements	Beyzanur Zeybek, Ali Eren Kurt	Completed	Statistical and visualization requirements established
T6	Defining visualization dashboard functionalities	Tunahan Oğuz, Alkım Doryan	Completed	Dashboard components and interactivity specified
T7	Documenting predictive modeling requirements	Beyzanur Zeybek, Ali Eren Kurt	Completed	Model selection, training, and validation requirements outlined
T8	Outlining automated reporting functionalities	Tunahan Oğuz, Beyzanur Zeybek	Completed	Report formats and export options defined
T9	Defining data export functionalities	Ali Eren Kurt, Alkım Doryan	Completed	Export formats and expected data structures outlined
T10	Defining query functionality requirements	Tunahan Oğuz, Beyzanur Zeybek	Completed	User query execution and filtering capabilities documented
T11	Establishing user access and role management	Alkım Doryan, Ali Eren Kurt	Completed	Authentication and authorization methods specified
T12	Documenting non-functional requirements	All team members	Completed	System quality and performance requirements detailed

	(performance, security, usability, etc.)			
T13	Writing and formatting the document	Beyzanur Zeybek, Alkım Doryan	Completed	Finalized structure and formatting completed
T14	Reviewing and finalizing the document	All team members	Completed	Final quality check and approval completed