League Management System

Analysis & Design

Project Manager: Izaak Ford-Dow

Date: 2025-02-012

# Approval Page

League Management Inc.

Leage Management System

LMS

This document serves to confirm the approval of the client, Leage Management Inc., and the project manager, Izaak Ford-Dow, for the outlined project. By signing below, both parties affirm that they have reviewed this document in full and agree to the proposed plan. Acceptance of this plan is confirmed by signing and dating below, allowing the project to move forward as detailed.

|  |  |  |  |
| --- | --- | --- | --- |
| Approver Title | Approver Name | Signature | Date |
| Client | Gray Lewis |  |  |
| Project Manager | Izaak Ford-Dow |  |  |

# Document Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Document Name | Date | Author(s) | Comments |
| 1.0 | CIS2261\_AD\_FordDow\_Izaak.docx | 2024/12/03 | Izaak Ford-Dow | First Draft |

Table of Contents

[Approval Page 1](#_Toc184339771)

[Document Tracking 2](#_Toc184339772)

[Management Summary 5](#_Toc184339773)

[System Overview 6](#_Toc184339774)

[System Statement 6](#_Toc184339775)

[System Deliverables / Assumptions / Constraints 7](#_Toc184339776)

[Deliverables: 7](#_Toc184339777)

[Non-Deliverables: 8](#_Toc184339778)

[Assumptions: 8](#_Toc184339779)

[Constraints: 8](#_Toc184339780)

[System Key Requirements 8](#_Toc184339781)

[System Modeling 10](#_Toc184339782)

[Event Table 10](#_Toc184339783)

[System Domain Class Diagram 13](#_Toc184339784)

[Information Management – Input Use Cases 13](#_Toc184339785)

[Reports – Output Use Cases 16](#_Toc184339786)

[System Components Details 19](#_Toc184339787)

[Program Design 19](#_Toc184339788)

[Design Class Diagram 19](#_Toc184339789)

[Information Management – Input Use Cases 20](#_Toc184339790)

[Reports – Output Use Cases 21](#_Toc184339791)

[Package Diagram 22](#_Toc184339792)

[Output Design 24](#_Toc184339793)

[Input Design 27](#_Toc184339794)

[Database Design 29](#_Toc184339795)

[Entity Relationship Diagram (ERD) 29](#_Toc184339796)

[Data Dictionary 30](#_Toc184339797)

[Support Processing Design 34](#_Toc184339798)

[Environmental Requirements 35](#_Toc184339799)

[Implementation Requirements 36](#_Toc184339800)

# Management Summary

This document outlines the system analysis and design process for the League Management System, a comprehensive platform designed to streamline the administration of sports leagues. The system enhances efficiency by automating league operations, ensuring seamless management of teams, schedules, and competition logistics. Key functionalities include:

* League and Division Management: Supports the creation and administration of leagues with multiple divisions, allowing for flexible competition structures.
* Team Management: Supports team creation, editing, and assignment to divisions (if applicable).
* Season Management: Enables creation of multiple seasons with custom start and end dates. Stores historical season data for reference and performance tracking.
* Role-Based Access Control: Implements a robust permission system, ensuring that administrators, coaches, and officials have appropriate access and control.
* Game Scheduling and Standings Tracking: Automates the scheduling of games, tracks scores and standings, and provides real-time updates for all participants.
* Team and Roster Management: Enables coaches to manage their teams, update rosters, and track players.
* Registration: Facilitates team and player registration, and reporting.
* Data Analytics and Reporting: Offers insights into league performance, team statistics, and trends for better decision-making.

**Key Recommendations:**

* Scalability and Performance: Ensure the system can accommodate multiple leagues and high user traffic.
* Web-Based Access: Provide 24/7 online access for administrators, teams, and players to manage league activities.
* Security and Compliance: Implement strong authentication, data encryption, and secure financial handling.
* User-Centered Design: Utilize iterative prototyping and stakeholder feedback to optimize usability and functionality.

This League Management System will enhance operational efficiency, improve user experience, and support league growth with a scalable and adaptable framework.

# System Overview

## System Statement

The proposed information system for the League Management System is designed to efficiently handle the administration of sports leagues. It provides a centralized platform for managing teams, seasons, schedules, and player statistics while ensuring seamless communication between administrators, coaches, and participants. Details of each module/subsystem are as follows:

**League, Division, and Season Management**

The system will allow administrators to create and manage leagues, and divisions, defining competition formats and rules. It will support the creation of multiple seasons, enabling structured play with configurable start and end dates. Historical data will be stored to track past league performance and trends.

**Team and Roster Management**

This subsystem will support team creation, editing, and assignment to divisions if applicable. Coaches and team managers will be able to update player rosters, manage player eligibility. Role-based permissions will ensure that only authorized users can modify team details.

**Game Scheduling and Standings**

The system will automate game scheduling based on predefined league formats such as round-robin or bracket play. It will track match results, update league standings in real time, and allow for rescheduling in case of cancellations. The standings module will calculate team rankings based on league rules.

**Registration**

This module will facilitate player and team registration, handling necessary documentation and eligibility verification.

**Analytics and Reporting**

League administrators will have access to analytical tools that provide insights into team and player performance, participation trends, and more. Reports will include detailed game statistics, and season comparisons to support data-driven decision-making.

This League Management System will streamline operations, improve participant experience, and provide a scalable solution for efficiently managing leagues of all sizes.

## System Deliverables / Assumptions / Constraints

### Deliverables:

* League and Season Management Module – Supports the creation and administration of leagues and seasons, including scheduling and rule configuration.
* Team and Roster Management Module – Enables team creation, roster updates, and player statistics tracking.
* Game Scheduling and Standings Module – Automates fixture generation, tracks results, and updates standings in real time.
* Registration – Manages player and team registration, financial reporting.
* Analytics and Reporting Module – Generates performance insights, statistical reports, and summaries for league administrators.
* User Documentation – Includes an installation guide, user manuals, and video demonstrations for system usage.

SQL Scripts – Provides scripts for database creation, schema management, and maintenance.

### Non-Deliverables:

League Management System Inc. will provide its own hardware and network infrastructure to be used with the LMS system.

### Assumptions:

* Adequate hardware and network infrastructure will be provided to support the system.
* Stakeholders will be available to provide input and review progress during development.
* Project funding and scope will remain consistent as initially agreed.

### Constraints:

* Timeline: The project must be completed within the agreed timeframe.
* Budget: Changes to scope, budget, or timeline require approval from the project manager and stakeholders.
* Technology Stack: The system must provide 24/7 online access for members and support all essential supercenter operations.

# System Key Requirements

**Inputs:**

* League details (name, structure, and competition rules)
* Team and player registrations
* Game schedules and match results

**Outputs:**

* League standings and match results
* System-generated reports (team and player statistics)
* Custom performance analysis
* Registration confirmations

**Data:**

* League and season configurations
* Team rosters and player information
* Game schedules, scores, and standings
* Report records

**Processes:**

* Automatic generation of schedules and standings updates based on match results
* Real-time tracking of player statistics and team performance
* Automated processing of team and player registrations
* Regular review of league performance reports

**Security:**

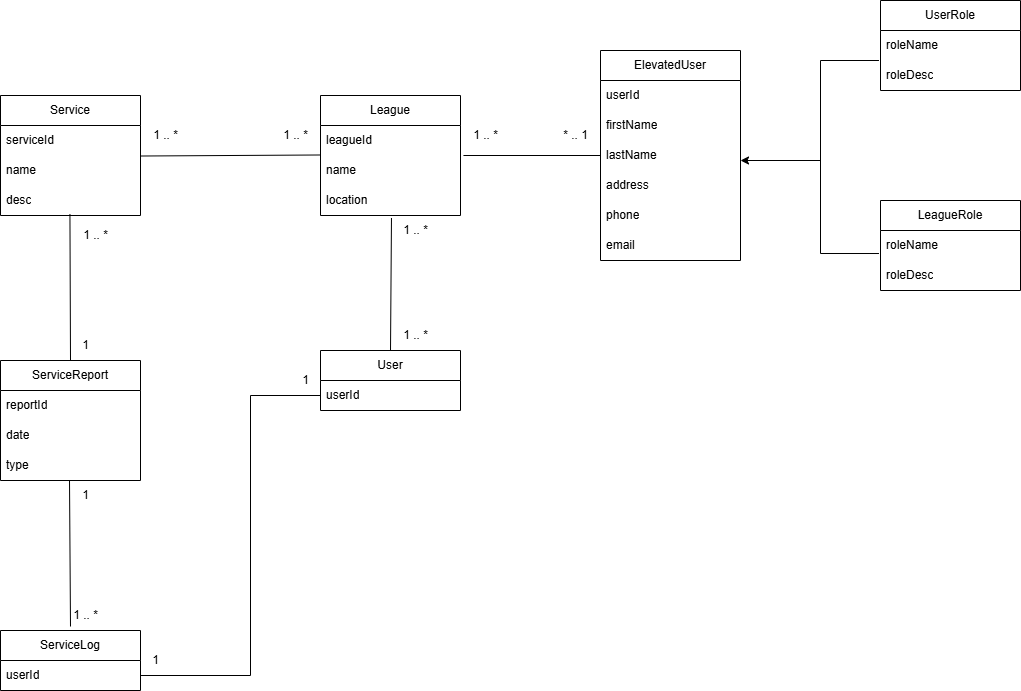
* User authentication for all system access:
  + Role-based access control for administrators, coaches, and more
  + Secure login with username and password, with optional multifactor authentication
* Standardized security protocols for data usage, transmission, and storage:
  + Encrypted HTTPS communication for secure web access
  + Secure storage of sensitive player, team, and financial data with encryption
  + Strict access control measures for managing league-related data

## System Modeling

### Event Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **League Management System Event Table** | | | | | |
| **Event** | **Trigger** | **Source** | **Use Case** | **Response** | **Destination** |
| Team Registration | A new team submits a registration request | Team Owner | Register a Team | Create new team entry in the system | Team Records |
| Season Creation | League Admin Creates a Season | League Admin | Create a new Season | Create a new season entry in the system | Season Records |
| Schedule Game | League Admin sets a game date | League Admin | Create a new Match | Create a new match entry in the system | Match Records |
| Game Submitted | User with appropriate permissions submits game results | User (appropriate permissions) | Create Match Result | Create a new Match Result entry in the system | Match Result records |
| Player Creation | User with appropriate permission submits a new player | User (appropriate permissions) | Create Player | Create a new player | Player Records |
| Create User | A request is made to sign up for the app | User | Create User | Create a new User | User Records |
| Create Role | Web Admin creates a new role | Web Admin | Create Role | Create a new Role | Role Records |
| Create Permission | Web Admin creates a new permission | Web Admin | Create Permission | Create a new Permission | Permission Records |
| Assign Permission to Role | Web Admin assigns permission to role | Web Admin | Link permission to role | Assigns permission to a role | Role Permission records |
| Assign a User to Role | Web admin assigns a User to s role | Web Admin | Link a user to role | Assigns role to user | User Role records |
| Create League Role | League Admin creates a new role | League Admin | Create League Role | Create a new League Role | League Role Records |
| Create League Permission | League Admin creates a new permission | League Admin | Create League Permission | Create a new League Permission | League Permission Records |
| Assign League Permission to Role | League Admin assigns permission to role | League Admin | Link League permission to role | Assigns League permission to a role | League Role Permission records |
| Assign a User to League Role | League admin assigns a User to s role | League Admin | Link a user to role | Assigns League role to user | League User Role records |
| Create a League News post | User with permissions | User with permission | Create League News Post | Creates a League News Post | League News records |
| Create a Team News Post | User with permissions | User with permission | Create Team News Post | Creates a Team News Post | Team News records |
| Create a Team Staff Member | User with permission | User with permission | Create Team Staff Member | Creates a team staff member | Team Staff records |

### System Domain Class Diagram



### Information Management – Input Use Cases

League Admin Creates a new Season:

The “League Admin Creates a new Season” use case occurs when a league admin wants to create a new season for their league. All users and staff will be able to view the details of the season.

*Use Case Diagram*

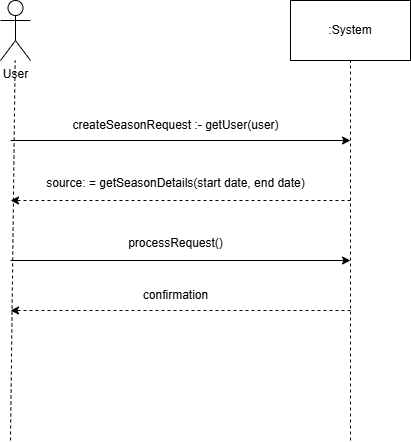
*A diagram of a season

AI-generated content may be incorrect.*

*Use Case Detailed Description*

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | League Admin Creates a new Season | |
| **Scenario:** | A league administrator uses the web app to create a new season for their league, allowing teams to participate and schedules to be generated. | |
| **Triggering Event:** | League admin logs into the system and initiates a request to create a new season. | |
| **Brief description:** | A league administrator wants to create a new season for an existing league. The system will allow them to define season settings, such as start and end dates, divisions, and rules. | |
| **Actors:** | Primary: League Admin, Secondary: System | |
| **Related use cases:** | Assign Teams to a Season, Generate Schedule | |
| **Stakeholders:** | * League Admin: Needs to create and manage the season. * Teams & Coaches: Need a valid season to participate in matches. * System: Needs to track season-specific stats and schedules. | |
| **Preconditions:** | * The league exists in the system. * The user is authenticated as a League Admin. * The previous season has ended (if applicable). | |
| **Postconditions:** | * The new season is recorded in the system. * Teams can be assigned to the new season. * League settings and custom stat tracking for the season are configured. | |
| **Flow of activities** | **Actor** | **System** |
| 1. Logs into the web app. | |  | | --- | | 1.1 Authenticates admin and displays the dashboard. |  |  | | --- | |  | |
| |  | | --- | | 2. Navigates to the "Seasons" section and selects "Create New Season". |  |  | | --- | |  | | |  | | --- | | 2.1 Displays the season creation form. |  |  | | --- | |  | |
| |  | | --- | | 3. Enters season details (name, start date, end date, rules, divisions). |  |  | | --- | |  | | 3.1 Validates input data. |
| |  | | --- | | 4. Confirms season creation. |  |  | | --- | |  | | |  | | --- | | 3.2 Saves season data to the database. |  |  | | --- | |  | |
|  | |  | | --- | | 3.3 Displays confirmation message and redirects admin to season management. |  |  | | --- | |  | |
| **Exception conditions:** | * Admin enters invalid season details (e.g., missing required fields). * System experiences an error while saving the season. * The league is not eligible for a new season (e.g., active season rules prevent it). | |

*Use Case Sequence Diagram*

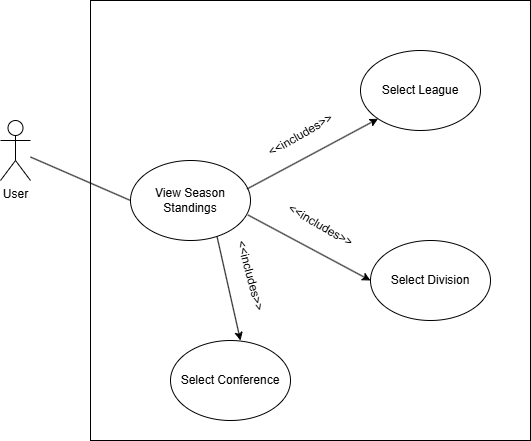
**

### Reports – Output Use Cases

User Views Season Standings

The “User Views Season Standings” use case occurs when a user requests to view the standings for a league. Every time a game is played, the standing for the league is updated and it will be processed to the app and available for viewing. A user will be able to view via multiple filters such as by division, conference, etc.

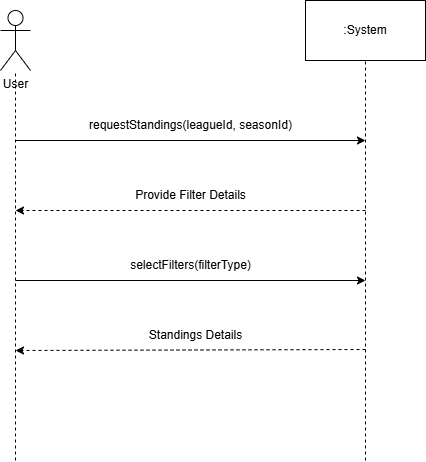
*Use Case Diagram*



*Use Case Detailed Description*

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | User Views Season Standings | |
| **Scenario:** | User uses the web app to view leagues season standings. | |
| **Triggering Event:** | User logs initiate a new view season standings request. | |
| **Brief description:** | A member wants to check the season standings of a current league. | |
| **Actors:** | User (primary), System (secondary) | |
| **Stakeholders:** | - User: Wants to view season standings.  - League Management Inc: Provides reports to keep customers engaged and happy. | |
| **Preconditions:** | - Season is in the system.  - League is in the system | |
| **Postconditions:** | - User receives report and can view standings. | |
| **Flow of activities** | **Actor** | **System** |
| 1. User opens the web app. | 1.1 Token is given to user. |
| 2. Customer navigates to a league then to the “View Standings” section. | 2.1 System presents filter options to the user. |
| 3. User selects filter such as division, conference. | 3.1 System validates the details entered (e.g., checks to ensure the user has data for the report). |
|  | 4. System displays report (log) to user. |
| **Exception conditions:** | - Invalid filter selection (e.g., no data to show).  - System is currently down. | |

*Use Case Sequence Diagram*

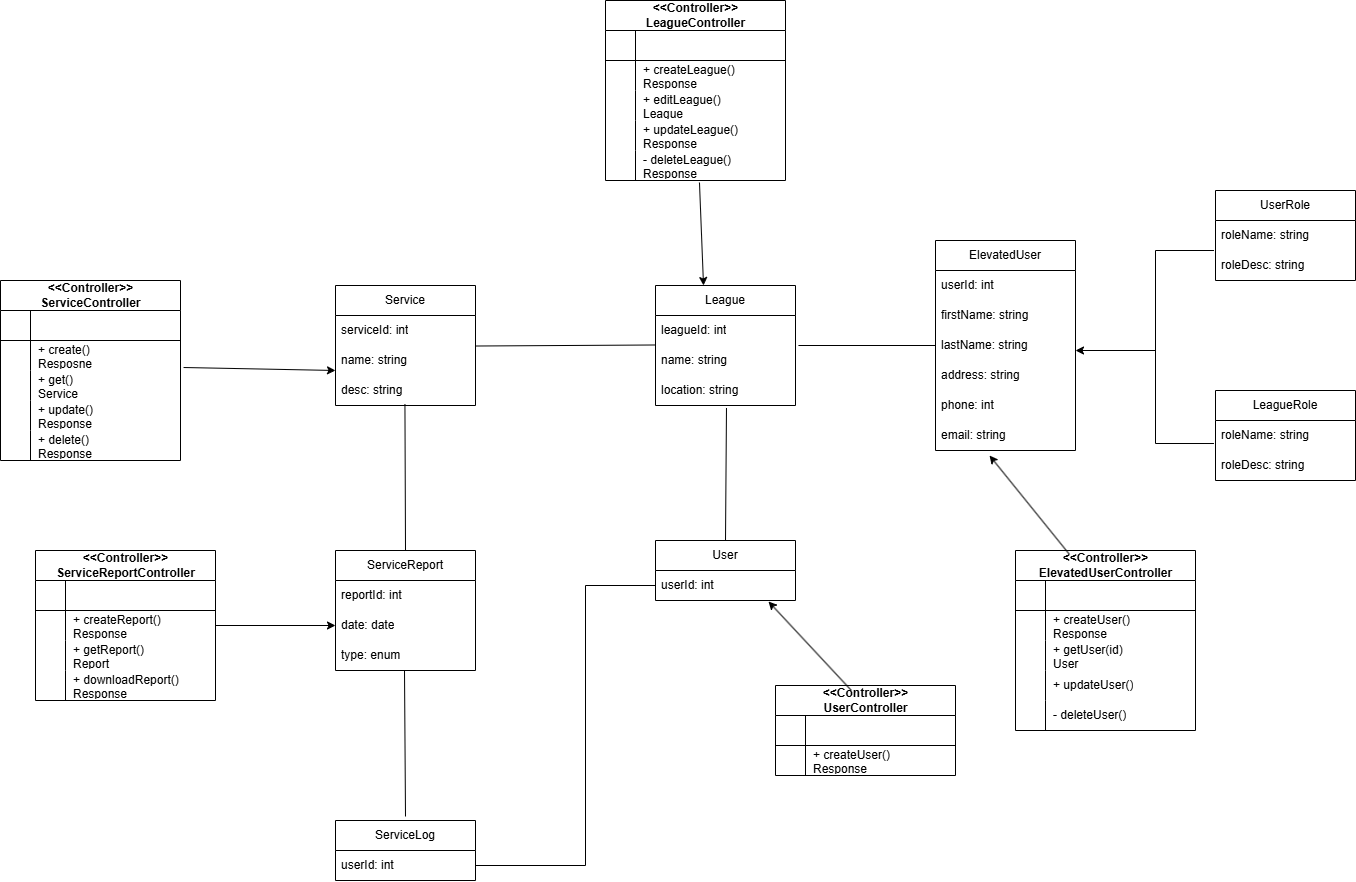


# System Components Details

## Program Design

### Design Class Diagram

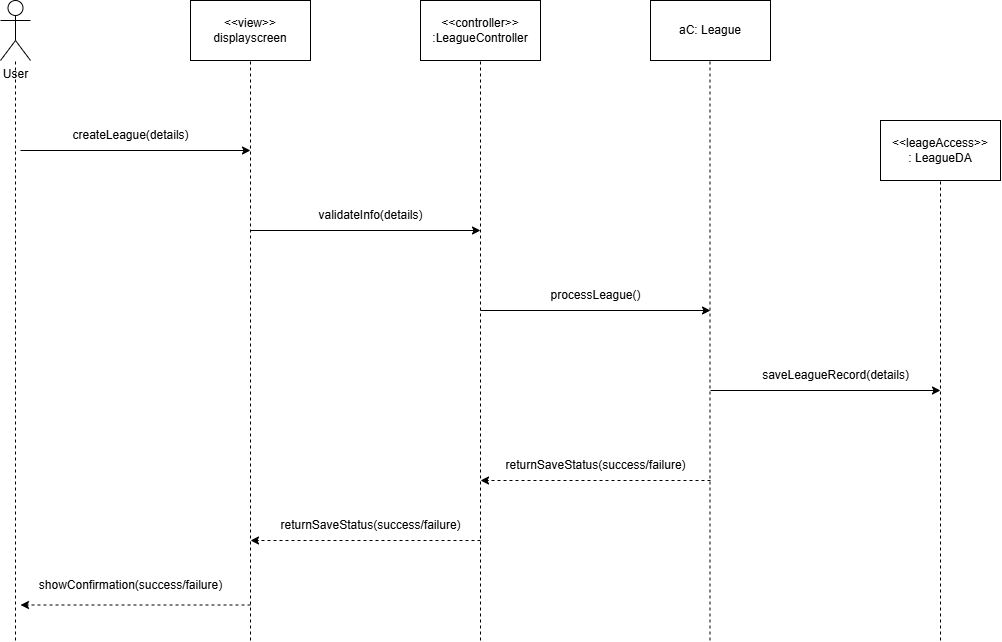
The Design Class Diagram on the following page demonstrates the object-oriented design approach for the League Management Inc. system. It outlines the data attributes, methods, and controller objects essential for the system's implementation. This diagram serves as an extension of the domain class diagram, focusing on the representation of software classes in the new system. The primary distinction is the inclusion of method signatures for each class, which detail the method names, input parameters, and return types in the implementation.



### Information Management – Input Use Cases

Create a League:

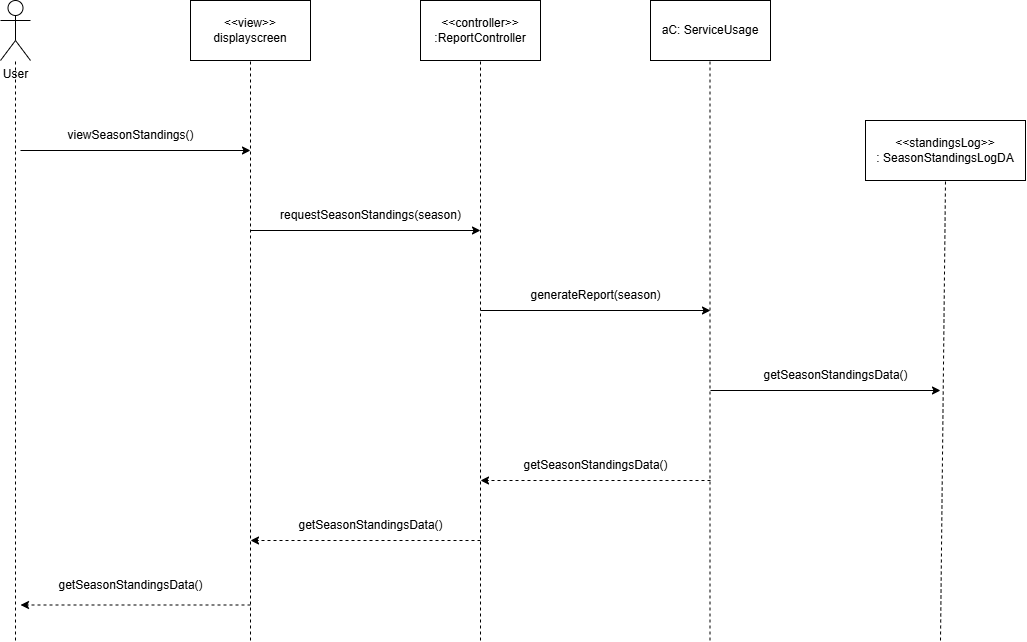
This sequence diagram illustrates the system interaction when a user creates a league. The member initiates a new league request, and the system processes the league information entered. Finally, the system returns a confirmation to the user at the end of the sequence.



### Reports – Output Use Cases

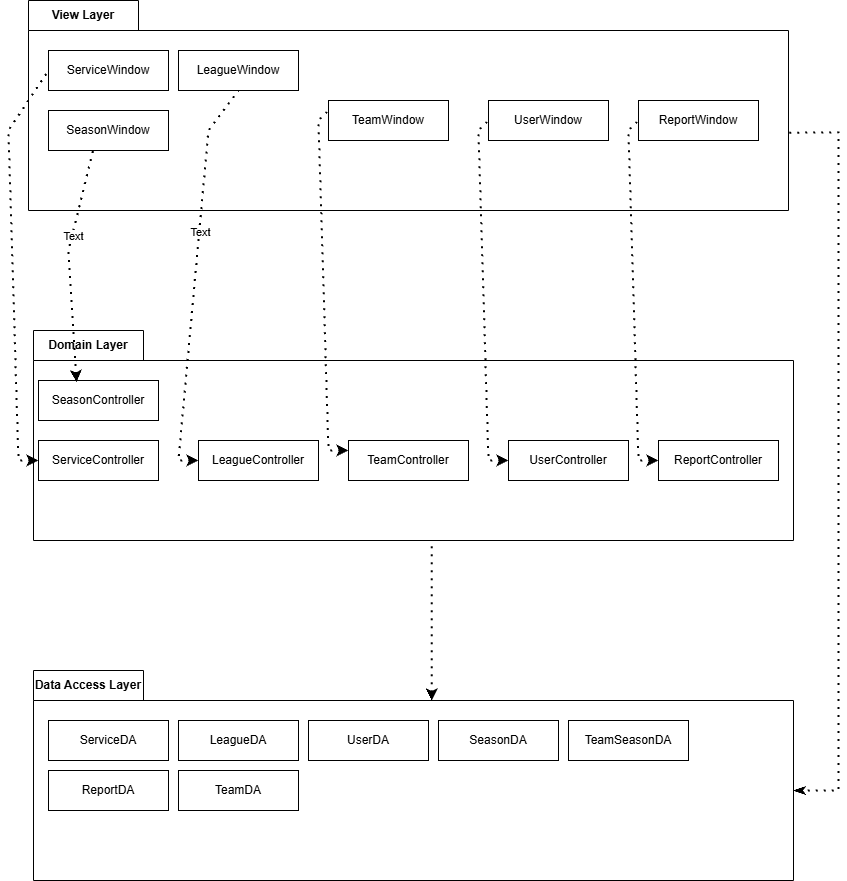
Request Season Standings Log:

This sequence diagram illustrates the system interaction when a user requests a season standings log. The system retrieves the season data from the database, generates the requested standings log, and displays it to the user for review.



### Package Diagram

This package diagram illustrates the relationships between various components of the system across the view, domain, and data access layers.



## Output Design

**System Documentation**

**Name of System:** League Management Inc.

**Date:** 2025-02-12

**Purpose:** Standings Report

|  |  |  |
| --- | --- | --- |
| FIELD | FIELD TYPE | FIELD LENGTH |
| TEAM NAME | INT | 50 |
| WINS | INT | 5 |
| LOSSES | INT | 5 |
| TIES | INT | 5 |
| GOALS FOR | INT | 5 |
| GOALS AGAINST | INT | 5 |

#### Comments:

1. Contains a log of standings for a league, along with the goals for and goals against for each team.
2. Date of the report is displayed in YYYY-MM-DD format.
3. Each log entry will have a service name (e.g., "Standings", "Player", etc.), the number of wins, losses, ties, goals against, and goals for, for each team.
4. Page number is shown in the bottom right corner.

#### Sort Sequence:

The standing log entries are sorted by Wins, with the team with the most appearing at the top.

#### Totals Required:

* Number of wins.
* Number of losses.
* Number of ties.
* Number of goals for.
* Number of goals against.

**Media:**

This report can be displayed on a screen for league admins or site admins to view. The report can also be printed on single-ply, standard white stock paper.

**Frequency:**

Generated monthly or on demand by the system for both league admins and site admins.

#### Distribution:

The report is available to league admins. Management may also have access.

#### Attachments:

Mock-up below.

**Standings Log**

League Management System

DATE: 2025-02-12

PAGE 1 OF 1

**LEAGUE STANDINGS REPORT**

League: Hockey PEI

Division: AAA

Month: FEBUARY 2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Team Name | Wins | Losses | Ties | Goals For | Goals Against |
| Sherwood Falcons | 14 | 2 | 5 | 88 | 35 |
| Charlottetown Abbies | 12 | 4 | 5 | 77 | 44 |
| Pownal Red Devils | 11 | 7 | 3 | 78 | 48 |
| Mid Isle Matrix | 10 | 10 | 0 | 70 | 77 |
| Tignish Aces | 9 | 12 | 2 | 50 | 78 |

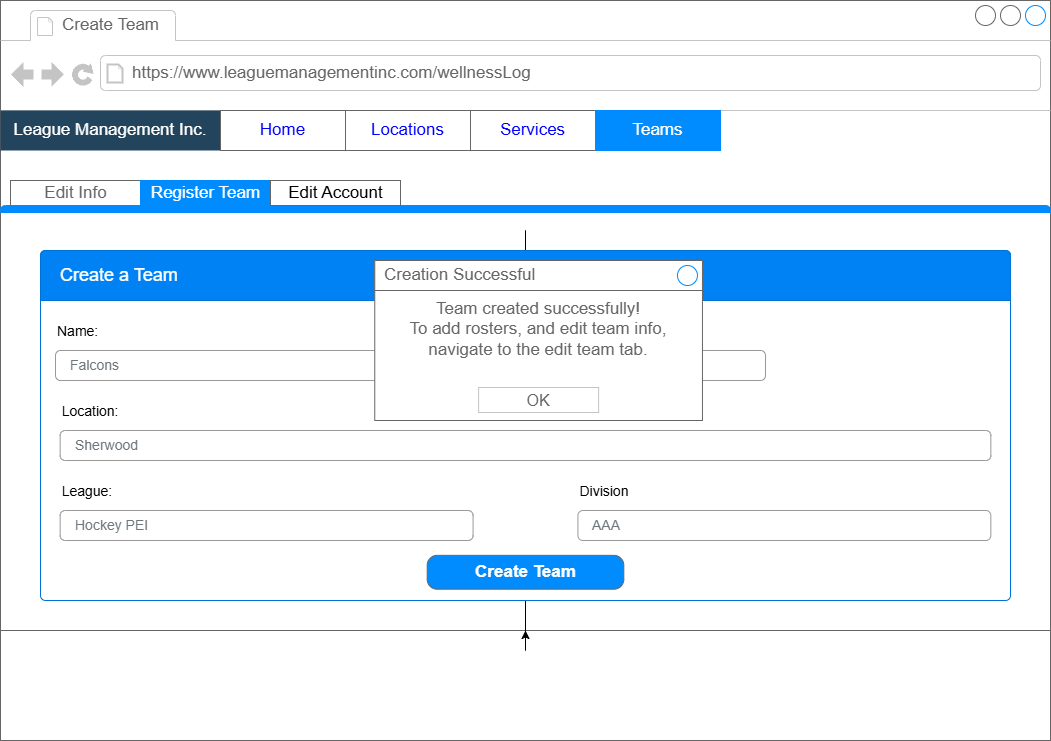
REPORT GENERATED ON: 2024-12-05

REPORT RANGE: 2025-01-02 to 2025-28-02

## Input Design

Create a Team:

This GUI design demonstrates how a member would create a new team in a league on the system. Once logged in, the member can navigate to the "Teams" section to view a create a team form. The form will require the member to input team details such as the name, location, and any additional information necessary for processing. After the fields have been validated, the member can submit the team by clicking a button. Upon successful submission, a confirmation pop-up will appear, notifying the member that the team was created successfully and providing instructions on how to modify the team further.



## Database Design

### Entity Relationship Diagram (ERD)

A computer screen shot of a computer

AI-generated content may be incorrect.

### Data Dictionary



### Support Processing Design

**Support Forms:**

* Team Registration and Consent Forms – Required for team participation in the league.
* Player Registration and Waiver Forms – Ensures legal compliance and player eligibility.
* Online Account Setup and Access Request Forms – Enables users to create and manage their accounts securely.

**Policy and Procedures:**

* Data Validation:
  + All team, player, and match data must be validated upon entry. Required fields must be completed, and data types must be verified to prevent invalid entries.
  + Sensitive data, including personal information, must be encrypted during transmission and storage.

**Output Management:**

* All system outputs, such as league standings, match results, reports, and player statistics, must be formatted consistently and include clear labels. Reports will adhere to a standard template approved by league administrators.
* Printed outputs must be securely disposed of using shredders, while electronic media must be sanitized using data erasure software. A disposal log, including dates, device serial numbers, and methods of disposal, will be maintained.

**Backup and Recovery:**

* Automated, cloud-based backups will be performed daily using a reliable service provider. Redundant local backups must also be scheduled weekly.
* A disaster recovery plan will be in place, with bi-annual backup testing to ensure data integrity and availability in case of system failure.

**System Access Control:**

* Only authorized personnel will have access to administrative features. Multi-factor authentication (MFA) will be implemented for staff accounts.

**Incident Reporting:**

* A reporting system will log and monitor any unauthorized access attempts or system failures, with regular audits to identify vulnerabilities and ensure compliance with security protocols.

# Environmental Requirements

#### Hardware:

* The server must meet specifications to support Spring Boot, React/Next.js**,** and MariaDB.

#### Software:

**Server:**

* Java version 21 or higher.
* MariaDB for database management.
* Apache for hosting the web application.
* Node.js 18 or higher

**Workstations:**

* Web interface compatibility with Google Chrome, Mozilla Firefox, or Microsoft Edge.
* Admin and data entry tools accessible through a modern web browser (no OS restrictions).
* The system is cross-platform compatible and can be accessed on Windows, MacOS, and Linux.

**Facility Additions/Modifications:**

* No facility additions or modifications are needed.

**Staff:**

* No additional staff required as this system will replace the current one.

# Implementation Requirements

**Data Entry / Conversion:**

* The league administration will oversee data migration from existing systems into the new League Management System database.
* Existing team, player, and match data must be parsed, migrated, and verified to ensure accuracy and integrity.

**Security:**

* Comprehensive user authentication mechanisms:
  + Secure login with username and password, with optional multifactor authentication for enhanced security.
* Adherence to robust security protocols for data usage, transmission, and storage:
  + Encrypted HTTPS web communications (TLS).
  + Encrypted database storage for sensitive information such as personal data and records.
  + Strict access controls, both physical and digital, to prevent unauthorized system access.
  + Role-based access control (RBAC) to define user permission levels for administrators, coaches, and players.

**Training:**

* All league administrators, coaches, and staff will require role-specific training on system functionalities, including:
  + Managing leagues, teams, and player rosters.
  + Scheduling matches and updating game results.
  + Handling registrations for leagues and players.
  + Accessing reports and analytics for league performance tracking.
  + Using the system within their assigned permissions.

**Post-Implementation Review:**

* A comprehensive review will be conducted one month after deployment to:
  + Assess whether all system requirements have been met.
  + Identify lessons learned for future system improvements.
  + Address any necessary bug fixes, optimizations, or feature enhancements based on user feedback.