

Software Requirements Specification

For

ATHLEDE - The Tournament Guide

12th of November 2024

Prepared by

Specialization	SAP ID	Roll No.	Name
AIML(NH)	500101976	R2142220195	Vaishnavi Dubey
AIML(NH)	500102209	R2142220125	Parth Soni
AIML(NH)	500105417	R2142220150	Rishi Raj Jain
AIML(NH)	500102177	R2142220061	Chitransh Soni



**Department of Artificial
Intelligence**

**School Of Computer
Science**

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand**

on 10th Nov 2024
12/11/2024

Table of Contents

Topic	Page No.
Table of Content	2
Revision History	3
1 Introduction	4
1.1 Purpose of the Project	4
1.2 Target Beneficiary	4
1.3 Project Scope	4
1.4 References	4
2 Project Description	5
2.1 Reference Algorithm	5
2.2 Data/ Data structure	5
2.3 SWOT Analysis	6
2.4 Project Features	8
2.5 User Classes and Characteristics	8
2.6 Design and Implementation Constraints	9
2.7 Design diagrams	9
2.8 Assumption and Dependencies	10
3 System Requirements	10
3.1 User Interface	10
3.2 Software Interface	10
3.3 Database Interface	10
3.4 Protocols	10
4 Non-functional Requirements	10
4.1 Performance requirements	10
4.2 Security requirements	10
4.3 Software Quality Attributes	11
5 Other Requirements	11

Revision History

Date	Change	Reason for Changes	Mentor Signature

1. INTRODUCTION

1.1 Purpose of the Project

Organizing college sports festivals has historically involved complex manual procedures that require substantial time, effort, and coordination. These challenges often lead to mismanagement,

scheduling conflicts, and inefficient use of resources. However, with the rise of digital solutions, there's a clear opportunity to streamline and enhance the planning of such events.

Athlead is designed to address these challenges by offering a comprehensive platform for automating key processes such as student registration, fixture creation, and court allocation. With

Athlead, participants and organizers alike benefit from an intuitive interface that provides seamless

registration and easy access to event details.

The platform automates critical tasks, reducing administrative burden and ensuring more precise

coordination. By integrating real-time updates and a dynamic interface, Athlead improves communication and information sharing among all stakeholders. Backed by a robust database management system (DBMS), the platform manages participant data, match results, and event schedules, providing an efficient and well-organized sports festival experience for all involved.

1.2 Target Beneficiary

Athlead primarily benefits event organizers, administrators, and participants by automating and streamlining sports event management through features like real-time scheduling, resource allocation, and data management. Its user-friendly interface improves the experience for participants, providing easy access to registration, schedules, and updates. With potential applicability in healthcare, Athlead can also support hospital administrators by efficiently managing appointments and resources. The platform's adaptable design makes it a valuable tool for any domain requiring organized and efficient scheduling and resource coordination.

1.3 Project Scope

Athlead is a versatile scheduling application designed to streamline resource management and appointment scheduling in high-demand environments like healthcare and large-scale events. Inspired by efficient healthcare systems such as Israel's, Athlead addresses challenges like long wait times and scheduling conflicts by providing real-time updates, automated scheduling, and resource allocation.

Key features include:

1. Automated Scheduling and Resource Allocation: Reduces manual intervention, optimizing service efficiency.

2. Real-Time Updates: Minimizes delays and conflicts by dynamically managing appointments and resources.

3. Robust DBMS Integration: Stores and organizes data for reliable access and decision-making.
4. User-Friendly Interface: Ensures easy access for both administrators and users.

Athlead is adaptable to any domain where efficient scheduling is essential, making it a valuable tool in healthcare and beyond.

1.4 References

1. Krotee, M., & Bucher, C. (2007). *Management of Physical Education and Sport*. McGraw-Hill.
2. Ali, M., & Parmar, J. (2022). "Automated Fixture Scheduling in Sports Tournaments". *International Journal of Computer Applications*.
3. Smith, J., & Green, T. (2019). "Technological Innovations in Sports Event Management". *Journal of Sports Technology*.
4. Ram, S., & Mishra, P. (2020). "Comparative Study of Digital Solutions for University Sports Management". *International Journal of Sports Science*.

2. PROJECT DESCRIPTION

2.1 Reference Algorithm

The project compares several scheduling algorithms, including:

- **First-Come, First-Served (FCFS)**
- **Shortest Job First (SJF)**
- **Round Robin (RR)**
- **Priority Scheduling**
- **Multilevel Queue Scheduling**
- **Multilevel Feedback Queue Scheduling**
- **Earliest Deadline First (EDF)**
- **Least Recently Used (LRU)**
- **Optimal Scheduling**

These algorithms are analysed based on their applicability and performance in fixtures scheduling scenarios for sporting events scenarios in tournaments and other sports events.

2.2 Data/Data Structure

Here's how we will incorporate the mentioned data structures into ATHLEAD

1. Hash Trees

- **Usage:** Efficient and secure management of participant and match records.
- **Benefits:** Hash trees (or Merkle trees) provide a way to verify the integrity of data and

ensure secure record management. They are particularly useful for quickly verifying the consistency of participant information and match results.

2. Array Lists

- Usage: Dynamic storage and retrieval of participant lists, match schedules, and event records.
- Benefits: Array lists are useful for managing collections of data where the size can change dynamically. They allow for easy insertion, deletion, and access of records such as teams, matches, and event schedules.

3. Trees (e.g., AVL Trees, Binary Trees)

- Usage: Efficient searching, sorting, and management of records such as scores and rankings.
- Benefits:
 - AVL Trees: Self-balancing binary search trees that provide $O(\log n)$ time complexity for insertion, deletion, and search operations. They are ideal for maintaining sorted records such as team rankings or player statistics.
 - Binary Trees: Useful for hierarchical data organization and fast lookups. They can be used for tasks like managing tournament brackets or hierarchical team structures.

2.3 SWOT Analysis

CPP-Based fixture making for tournaments and other sports events

Strengths:

1. Streamlined Event Management:

Athlead simplifies the organization of sports festivals by automating tasks like student registration, fixture creation, and court allocation, reducing manual workload.

2. Real-Time Data Handling:

The app's use of a database management system (DBMS) ensures real-time updates and efficient data handling, from participant information to match results, enhancing overall event coordination.

3. User-Friendly Interface:

The intuitive design makes it easy for both organizers and participants to navigate the app, improving user engagement and ensuring a smooth experience.

4. Scalability:

Athlead can be expanded to accommodate more sports or additional event features, making it adaptable for future use cases.

5. Efficiency and Accuracy:

Automated scheduling and slot allocation minimize the chances of human error, ensuring that the event runs smoothly and according to plan.

6. Cross-Platform Compatibility:

The app can be adapted for mobile and web platforms, making it accessible to a broad range of users.

Weaknesses:

1. Limited Customization:

Athlead may lack customization options for event organizers who want to tailor the app to specific sports or rules, which could limit its appeal for certain types of events.

2. Dependency on Digital Infrastructure:

The app's effectiveness depends on reliable internet and server connectivity. Any downtime could disrupt the management of the event.

3. Initial Learning Curve:

Users unfamiliar with digital sports management platforms may experience a learning curve when first using Athlead, potentially delaying event setup.

4. Limited Feature Set:

In its initial phase, Athlead may focus primarily on core management tasks, potentially missing advanced features like live score updates, streaming, or audience engagement tools.

Opportunities:

1. Widespread Adoption in Colleges:

With increasing digitization of events, Athlead could become the go-to app for managing not only sports fests but also other types of college events and competitions.

2. Integration with External Platforms:

Athlead can expand by integrating with other platforms like social media for participant engagement or fitness tracking apps to provide a more holistic event experience.

3. Event Analytics:

By leveraging the stored data, Athlead could provide detailed analytics on performance trends, participation rates, and resource utilization, which can offer valuable insights for organizers.

4. Customizable Add-ons:

Offering customizable features, such as personalized event branding or tailored event rules, could increase Athlead's appeal to a wider variety of institutions.

Threats:

1. Competing Event Management Tools:

There are other event management apps, some of which may offer more specialized or advanced features, making it harder for Athlead to establish a foothold in the market.

2. Data Security Risks:

Handling sensitive participant data, such as personal information and event results, poses potential security risks. Ensuring robust data protection measures is essential to maintain user trust.

3. User Expectations:

Event organizers and participants may expect more advanced features like live streaming, real-time scoreboards, or social engagement tools, which could challenge the app's initial offering.

4. Technical Issues During Events:

Any technical glitch or failure during a live sports fest could severely impact user experience and damage the app's reputation, leading to reluctance in future adoption.

2.4 Project Features

Implementation of Multiple Scheduling Algorithms

- Round Robin, Priority Scheduling, Earliest Deadline First for match slot allocation based on game type, participant count, and resource availability.

Performance Evaluation of Scheduling Algorithms

- Metrics such as wait time, resource utilization, and slot assignment fairness to assess efficiency and event flow.

Comparative Analysis & Optimal Algorithm Recommendation

- Data-driven comparison of algorithm performance and adaptive AI recommendations for different sports, teams, and resource constraints.

Optimal Scheduling for High-Impact Scenarios

- Advanced strategies for handling complex slot management, minimizing conflicts, maximizing resource use, and supporting large-scale events.

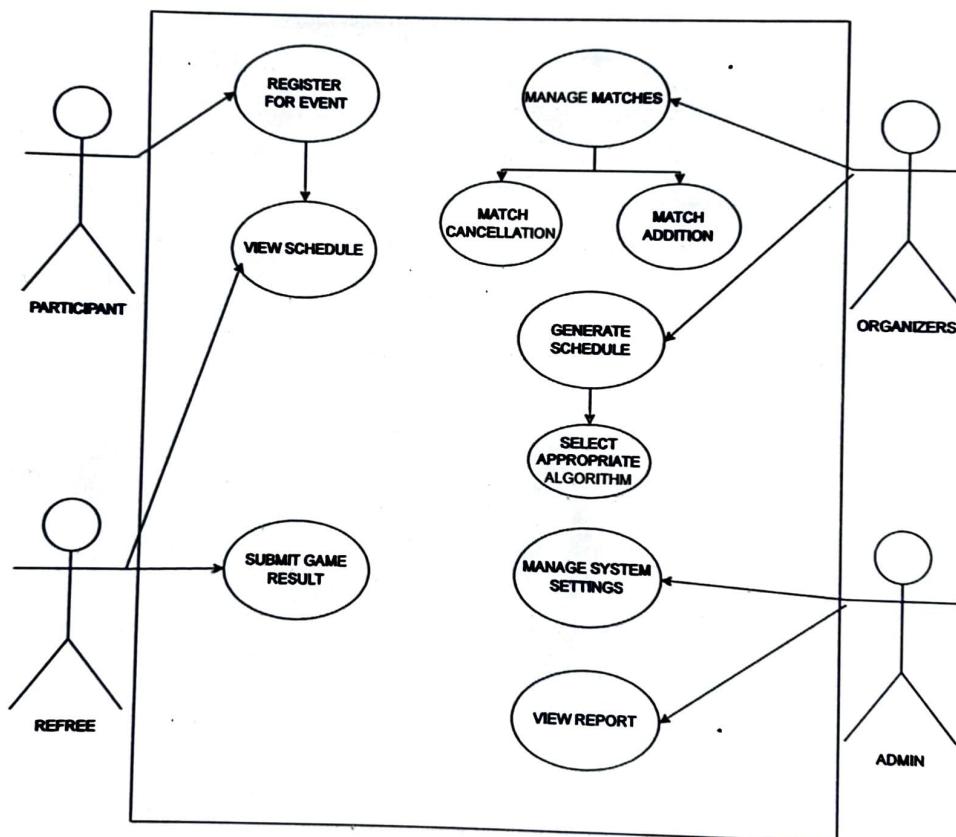
2.5 User Classes and Characteristics

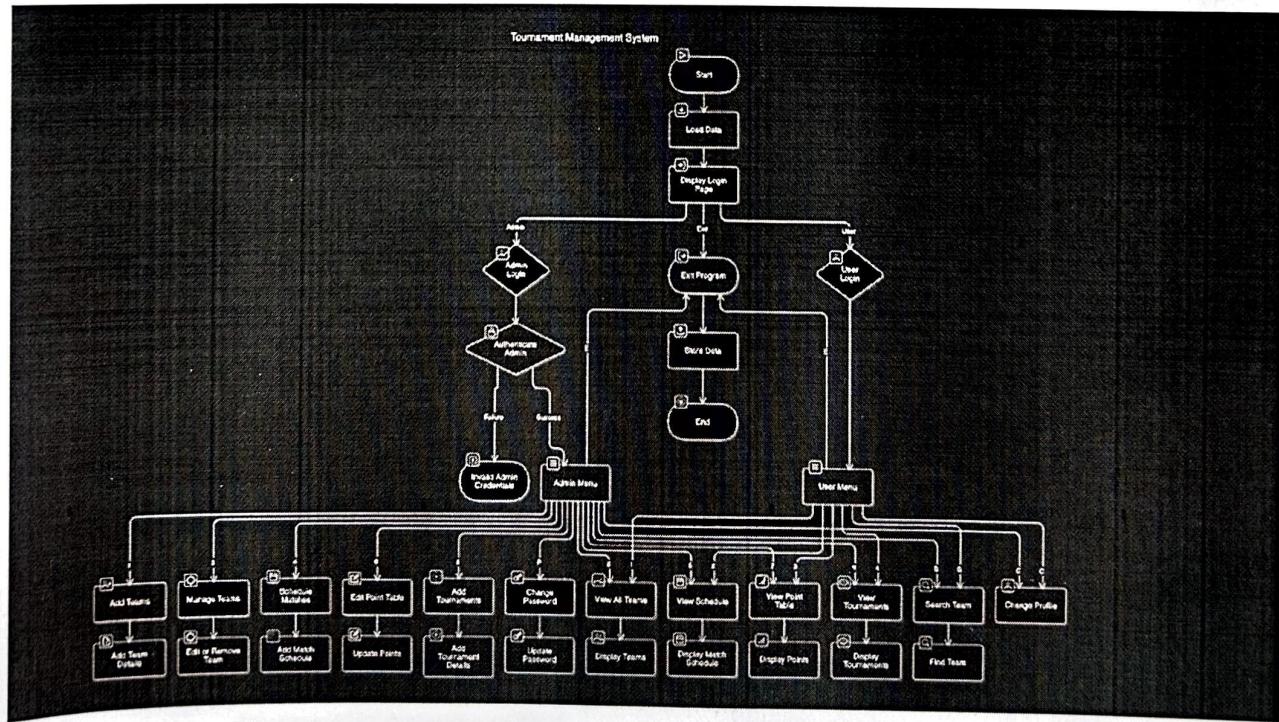
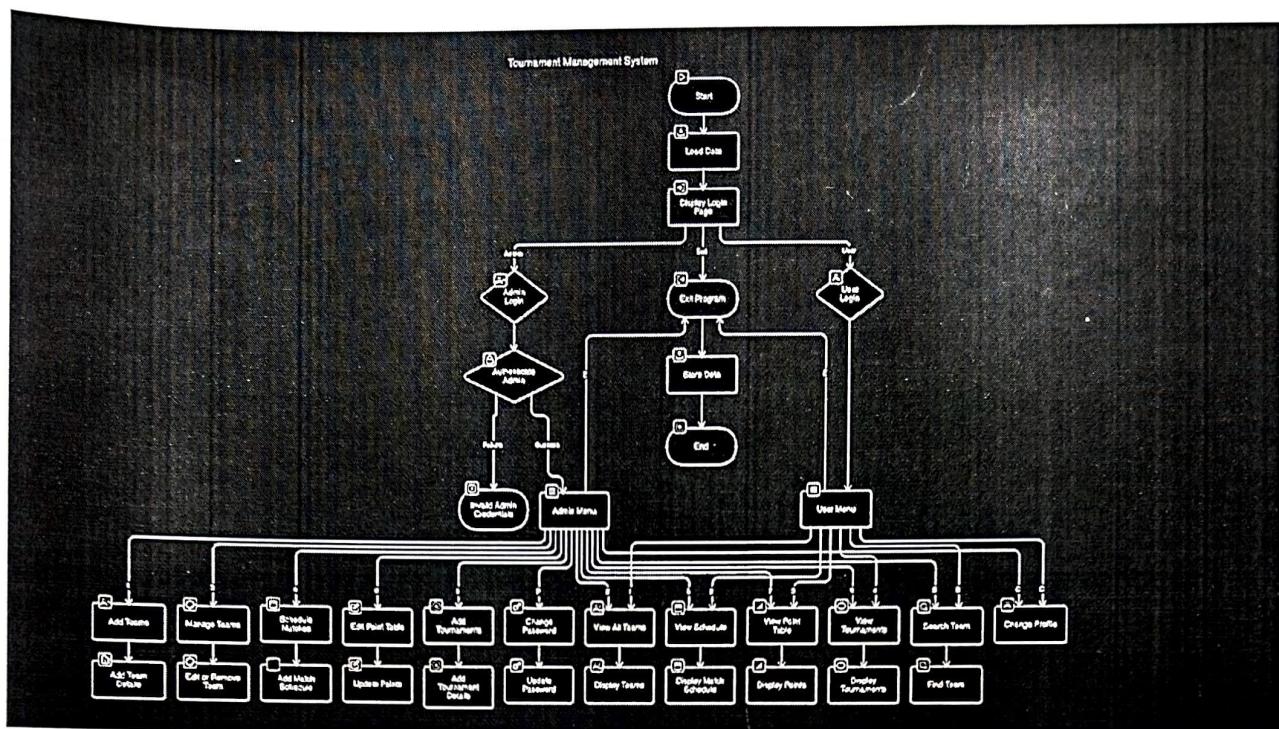
- Participant-A college student who registers for events, views schedules, and receives notifications for match timings and results.
- Organizer-A staff or student member responsible for managing game fixtures, assigning courts, and ensuring matches proceed as scheduled.
- Referee-An individual assigned to oversee matches, track game results, and submit performance data for participants.
- Admin-A user with access to system-wide controls, including managing event settings, user roles, schedules, and generating comprehensive event reports.

2.6 Design and Implementation Constraints

- Complexity: Managing diverse scheduling algorithms and scenarios for different sports, match types, and participant numbers within a single system.
- Scalability: Ensuring the platform can handle high volumes of registrations, fixture changes, and game schedules as participant numbers grow.
- Real-time Processing: Achieving timely slot allocation and scheduling updates in a real-time environment, especially during peak event hours.

2.7 Design Diagram – Use case





2.8 Assumptions and Dependencies

Assumptions

- Accurate data on participant registrations, event details, and resource availability will be readily available.
- The system can simulate and test various scheduling scenarios to recommend optimal solutions for different sports events.

Dependencies

- C++ programming environment and necessary libraries for implementing scheduling algorithms and real-time processing.
- Database system for storing participant data, schedules, and event outcomes, along with tools for data analysis and reporting.

3. SYSTEM REQUIREMENTS

3.1 User Interface

- A simple, user-friendly interface that allows users to register for events, view schedules, and receive match notifications.

3.2 Software Interface

- Integration with C++ libraries to implement scheduling algorithms.
- Compatibility with tools to simulate and analyze various scheduling scenarios for optimized event management.

3.3 Database Interface

- A database system to store and manage participant registrations, game schedules, and event outcomes.
- Efficient data retrieval and updates to support real-time scheduling adjustments.

3.4 Protocols

- Use of standard communication protocols to facilitate data exchange between the user interface and the scheduling system.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements

- The system should process and allocate match slots promptly, minimizing wait times for users.
- The scheduling engine should handle high volumes of registration and scheduling requests without significant performance degradation.

4.2 Security Requirements

- The system must securely store and transmit participant and event data, ensuring protection against unauthorized access and data breaches.

4.3 Software Quality Attributes

- **Reliability:** The system should consistently deliver accurate scheduling results and maintain stability during high-traffic periods.
- **Usability:** The user interface should be intuitive and accessible for all users, regardless of technical expertise.
- **Maintainability:** The codebase should be modular, well-documented, and structured to facilitate future modifications and enhancements.

5. OTHER REQUIREMENTS

- The system should support adaptable scheduling configurations, enabling customization of algorithm parameters to suit different event needs.
 - Future updates could include machine learning-based algorithms that continuously improve scheduling efficiency and adaptability based on historical data and usage patterns.
-