Ca' Foscari University of Venice

Department of Environmental Sciences, Informatics and Statistics

Course Title : SOFTWARE ARCHITECTURES
Course Code: CM0639-1
Assignment: Task 1 & Task 2

Prepared by:
Student Name: Yared Debela
Student Id: 913882
Instructor : Prof. Pietro Ferrara
Academic Year: 2025-2026

Submission Date: 02-01-2026

Name: Yared Debela Id: 913882

Task 1 – System Architecture Design

Team-Based Soccer Tournament Management System

1. Introduction

This project focuses on the design and implementation of an IT system for managing team-based soccer tournaments. The system supports the complete lifecycle of a tournament, including team and player registration, match pairing and scheduling, match event tracking, result recording, standings calculation, and public dissemination of tournament information. To enable architectural comparison, the system is implemented using two different architectural approaches:

- A monolithic architecture
- A distributed (service-based) architecture

Both implementations satisfy the same functional and non-functional requirements. This allows an evaluation of architectural trade-offs related to scalability, performance, reliability, deployability, maintainability, and cost.

2. Scope and Domain Definition

2.1 Sport and Tournament Type

- Selected sport: Soccer
- Category: Team-based sport
- Tournament format: League-style tournament organized in rounds

Soccer was selected due to its widespread familiarity and well-defined structure involving teams, matches, results, standings, and strong public interest. Although the system is focused on soccer, the domain model is designed to be extensible to other team-based sports.

3. Stakeholders and Users

3.1 Stakeholders

The primary stakeholders involved in the system are:

- Tournament organizers, who define operational requirements
- Course instructors, who evaluate the architecture and implementation
- The development team, responsible for system design and implementation

3.2 User Roles and Expected Scale

The system supports the following user roles:

- Administrator
  - Manages tournaments, venues, teams, matches, results, and standings
  - Estimated users: 15–20
- Coach / Team Manager
  - Manages team information and player rosters
  - Estimated users: Up to 100
- Referee / Match Official
  - Records match events and submits official match results
  - Estimated users: Approximately 50
- Spectator (Public User)
  - Accesses tournament information through a public, read-only website
  - Estimated users: Up to 30,000–50,000 concurrent users during peak events

4. Functional Requirements

The following functional requirements are implemented in both the monolithic and distributed systems:

- FR-1: Team Enrollment

  Teams can be registered for tournaments with basic team information.

- FR-2: Player Management

  Teams can register players with attributes such as name, position, and jersey number.

- FR-3: Match Pairing and Scheduling

  Matches are created, paired, scheduled, and assigned to venues and tournament rounds.

- FR-4: Match Event Tracking

  Referees can record match events such as goals, cards, and substitutions.

- FR-5: Match Results and Reports

  Final scores and official match reports are stored after each match.

- FR-6: Standings and Rankings

Tournament standings are automatically calculated based on match results.

- FR-7: Public Tournament Website

  Spectators can view tournaments, matches, results, standings, and match reports via a public web interface.

5. Non-Functional Requirements

The system satisfies the following non-functional requirements:

- NFR-1: Scalability

  The system must support large numbers of spectators and authenticated users during peak tournament periods.

- NFR-2: Performance

  Match events and results must be processed with low latency, and public pages must load quickly.

- NFR-3: Availability and Reliability

  Public tournament information must remain accessible during tournaments, and confirmed match data must not be lost.

- NFR-4: Maintainability and Modularity

  The system should be modular and easy to extend with new features, reporting capabilities, or tournament formats.

- NFR-5: Security

  Only authorized users (administrators, coaches, and referees) are allowed to modify tournament data, while spectators have read-only access.

- NFR-6: Deployability

  The system must be easy to deploy, reproduce across environments, and support updates with minimal downtime.

- NFR-7: Cost Efficiency

  The system should minimize operational costs by using open-source technologies and limiting infrastructure usage outside active tournament periods.

6. Architectural Alternatives

The system is implemented using two independent architectural solutions, each providing a complete realization of the same requirements:

- A monolithic architecture
- A distributed, service-based architecture

This approach enables a structured comparison of architectural quality attributes.

7. Monolithic System Architecture

In the monolithic implementation, the entire system is developed and deployed as a single application.

Characteristics

- Centralized business logic
- Single relational database
- Simple deployment and configuration
- Strong internal cohesion

Responsibilities

- User authentication and authorization
- Tournament, team, and player management
- Match scheduling and event tracking
- Result processing and standings calculation
- Rendering of administrative and public web interfaces

This architecture prioritizes simplicity, ease of development, and low operational overhead.

8. Distributed System Architecture

In the distributed implementation, the system is decomposed into independent services, each responsible for a specific domain concern.

Example Services

- Authentication and Authorization Service
- Tournament and Team Management Service
- Match Scheduling and Event Tracking Service
- Results and Standings Service
- Public Website Service

Characteristics

- Independent deployment and scaling of services
- REST-based communication between services
- Improved fault isolation
- Better support for scalability and availability during peak events

This architecture prioritizes modularity, scalability, and resilience.

9. Common Data Model

Both architectural approaches are based on the same logical data model, including:

- Tournaments, rounds, and venues
- Teams and players
- Matches, match events, and match reports
- Standings and rankings
- Users, roles, and permissions

While the logical data model is shared, the physical implementation differs between the monolithic and distributed systems.

10. Implementation Plan

Both systems are implemented using the following technologies:

- Laravel as the backed framework
- MySQL for relational data storage
- Docker for containerization and deployment

Each system is deployed independently and can be executed and evaluated separately.

11. Conclusion

This task defined the scope, requirements, and architectural context of a Team-Based Soccer Tournament Management System. The system supports multiple user roles, high spectator traffic, and critical tournament data, while remaining maintainable and cost-effective. Implementing both monolithic and distributed architectures enables a concrete comparison of architectural trade-offs while satisfying the defined functional and non-functional requirements.

Name: Yared Debela Id: 913882

Task 2 – Architecture Characteristics

Team-Based Soccer Tournament Management System

This document identifies and analyzes the architecture characteristics that are most relevant to the Team-Based Soccer Tournament Management System defined in Task 1. The selected characteristics are derived directly from the system's functional and non-functional requirements, user roles, and usage context. For each characteristic, measurable target levels are defined to support objective architectural evaluation.

1. Scalability

Why it is important:

The system must support a large number of concurrent spectators accessing public tournament information while simultaneously supporting authenticated users such as administrators, coaches, and referees during active match days.

Derived from Task 1:

- Section 3.2 – User Roles and Expected Scale
- FR-7: Public Tournament Website
- NFR-1: Scalability

Quantification:

- Support up to 30,000–50,000 concurrent spectator users accessing public pages.
- Support up to 2,000 registered players across all teams.
- Support up to:
  - 10–20 administrators
  - Up to 100 coaches/team managers
  - Approximately 50 referees and match officials
- Maintain an average response time of under 2 seconds for public read-only requests during peak load.

2. Elasticity

Why it is important:

System usage varies significantly across the tournament lifecycle, with strong traffic peaks during match days and final rounds and low usage outside tournament periods.

Derived from Task 1:

- Section 1 – Introduction
- Section 2.1 – Tournament Type
- NFR-1: Scalability

Quantification:

- Ability to scale system resources up or down within 5 minutes.
- Handle traffic bursts of up to 8–10× the average load without service degradation.
- Reduce infrastructure usage during off-peak periods to minimize operational cost.

3. Performance and Responsiveness

Why it is important:

Referees and administrators require fast feedback when recording match events and results, and spectators expect near real-time updates of scores and standings.

Derived from Task 1:

- FR-4: Match Event Tracking
- FR-5: Match Results and Reports
- FR-6: Standings and Rankings
- NFR-2: Performance

Quantification:

- Match event submission processed in under 1 second.
- Final match result submission processed in under 1 second.
- Public pages (matches, results, standings) load in under 2 seconds under normal conditions.
- Updated standings visible to spectators within 5 seconds after result submission.

4. Reliability

Why it is important:

Match results and standings represent official tournament data and must remain correct and available throughout the tournament. Data loss or inconsistencies would compromise the integrity of the competition.

Derived from Task 1:

- FR-5: Match Results and Reports
- FR-6: Standings and Rankings
- NFR-3: Availability and Reliability

Quantification:

- System availability of at least 99.5% during tournament periods.
- Zero loss of confirmed match results and match events.
- Automatic recovery from individual service failures in the distributed architecture.
- Consistent standings across all system views.

5. Deployability

Why it is important:

The system may require updates or fixes during active tournaments. Deployments must be fast, repeatable, and low-risk to avoid service disruption.

Derived from Task 1:

- NFR-6: Deployability
- Section 7 – Monolithic System Architecture
- Section 8 – Distributed System Architecture

Quantification:

- Deployment time of under 10 minutes.
- Support for zero-downtime deployments of public-facing components.
- Monolithic system deployable as a single containerized unit.
- Distributed services deployable independently.

6. Modularity

Why it is important:

The system is expected to evolve, potentially adding new reporting features, statistics, or support for additional team-based sports.

Derived from Task 1:

- NFR-4: Maintainability and Modularity
- Section 8 – Distributed System Architecture
- Section 9 – Common Data Model

Quantification:

- Clear separation between:
  - User authentication and authorization
  - Tournament and team management
  - Match scheduling and event tracking
  - Results and standings computation
  - Public website
- No cyclic dependencies between major modules or services.
- Each service in the distributed architecture owns its domain logic and data.

7. Security

Why it is important:

Only authorized users must be able to modify sensitive tournament data, while spectators must have read-only access to public information.

Derived from Task 1:

- Section 3.2 – User Roles and Expected Scale
- FR-4: Match Event Tracking
- FR-5: Match Results and Reports
- NFR-5: Security

Quantification:

- Role-based access control for:
  - Administrators
  - Coaches/Team Managers
  - Referees and Officials (≈50 users)
- 100% of write operations require authentication.
- Public website accessible without login and fully read-only.

8. Overall Cost

Why it is important:

The system is developed within an academic and institutional context with limited budget constraints and must remain cost-efficient to operate.

Derived from Task 1:

- Section 10 – Implementation Plan
- Section 11 – Conclusion

Quantification:

- Use of open-source technologies only (Laravel, MySQL, Docker).
- Minimal cloud or infrastructure usage outside tournament periods.
- Prefer shared resources for the monolithic system.
- Independent scaling of distributed services to reduce unnecessary cost.

Conclusion

The architecture characteristics defined in this task directly support the functional and non-functional requirements of the Team-Based Soccer Tournament Management System. By defining measurable targets for each characteristic, the system architecture can be objectively evaluated and compared across both the monolithic and distributed implementations, enabling informed architectural decision-making.