Storyline for Football Club Management System Project:

The Football Club Management System is built to simplify club operations by managing essential areas like player and staff data, game schedules, and performance statistics. Through a structured approach, it keeps all key information organized and accessible for club managers. Player profiles are created to capture important details, such as individual strengths, performance stats, and areas for improvement. This setup helps coaches and managers monitor each player's development and make informed decisions on team lineups. They can quickly set up teams, view player stats, and adjust lineups as needed, knowing the system is designed to handle updates and changes without disruptions.

This club relies on a management system built using the S.O.L.I.D. principles. These principles make sure each part of the system works on its own but fits together seamlessly with the rest.

Single Responsibility Principle (SRP)
Each class in the system has a single, specific job. For example, SeniorPlayer and JuniorPlayer classes manage different types of players, while Coach and Staff classes manage coaching and service. By keeping these roles separate, each part of the system stays focused on its unique job, making the code easier to maintain and understand.

Open/Closed Principle (OCP)
The system is designed to allow new features without changing existing code. For example, the Sponsor class can easily add new types of sponsorships through its extendContract method. Similarly, Player subclasses can be expanded to support new player needs without modifying the core structure.

Liskov Substitution Principle (LSP)
The system allows different player types (SeniorPlayer and JuniorPlayer) to be used interchangeably within teams. This means that both player types can be managed by the same Team class, making it easy to work with different types of players under one common structure.

Interface Segregation Principle (ISP)
Specific interfaces, such as Injurable, SalaryManageable, and RevenueManageable, are created so classes only include the methods they need. For instance, Player implements Injurable and SalaryManageable to handle injuries and salaries. Meanwhile, the Stadium class only uses RevenueManageable to focus on ticket sales and revenue, without unrelated features.

Dependency Inversion Principle (DIP)
The system's main components, like Club and Team, rely on general guidelines (interfaces) rather than specific class details. This setup makes it easy to add new roles or player types in the future without major changes. The system's design ensures that each part can work with others easily, while staying flexible.