FINAL YEAR PROJECT

South East Technological University

School of Science

Department of Computing and Mathematics

SpringBoot-based Football League Management platform system

Author: Supervisor:

Zhenyang Wan **Aiman Khalil**

Student ID: 20109227

BSc. (Hons) in Software Systems Practice

Contents

Table of Content

1. Introduction

1.1. Project Overview

In the context of the digital era, the organization of sports events, especially football matches, has become more and more complex, which requires a system to manage and ensure the events held successfully. Traditional way of managing tournaments often has some problems[1], including scheduling conflicts, communication inefficiencies, and manual data handling. To address these issues, our football competition management system provides a platform that facilitates coordination among players, teams, referees, and organizers. The system will have functionalities such as team registration, match scheduling, score tracking, notification systems. This solution not only helps to improve the management of sports events but also enhances the experience for participants by providing a one-stop platform.

1.2. Motivation

In order to solve the current challenges, Due to the lack of mature and convenient football match management system to manage the sports events, the participants of the matches can not get a good experience and the sponsor can not find a easier way to deal with unexpected changes in the event and collect the event information quickly. Therefore, it is particularly urgent and necessary to develop a system that can manage the football league and send notifications to those who concern about it. The system will provide a flexible management plan to deal with all the details about the football league and provide more user-friendly services.

1.3. Objectives

The target of the project is to build a reliable, scalable and user-friendly platform that makes the football match organizations better and improve the participants experience. We hope to achieve the following points: First, generate the match schedules automatically to avoid the conflict and make the schedule more reasonable. The second is to optimize the match information access process by sending notifications about the matches. The third is to assign roles for each user to improve the management and simplify the process of event registration. The fourth is to collect the result of matches and generate the ranking details to avoid human error.

1.4. Scope and Limitations

The project will be developed focusing on key functionalities, including match schedule generation, user notification, team and player management, results tracking. An agile development approach will be used during the development process, and keep iterative optimization. However, considering the time and budget constraints, we will not realize the features such as live match and integration of external sports platform. And original system may focus on school football matches. In addition, we will focus on the safety and performance to ensure the system's reliability and room for future improvement

2. Feasibility Study

2.1. Market Analysis

Now, with the government and sport organizations supporting campus football matches[2], there are more and more football matches on campus which creates a huge demand in football match management system. Because the traditional way of organizing a match may cost a lot, and all the campus football leagues are looking for digital solution to improve the management and store data into a database. In this situation, the market needs a system to solve the problem. Using a system can remarkably increase the efficiency of management and reduce the cost.

2.2. Application Scenarios

Our System is designed with a wide range of application scenarios, ensuring it meets the needs of all the user groups. For instance, students can easily register their teams and participate in tournaments. The organizers in schools and universities can use the system to check match scheduling, track match results, and handle emergencies. Additionally, the platform provides tools for organizing events and registering participants. These scenarios prove the system can offer flexible, user-friendly and customized platform for organizing football matches in campus.

2.3. Brand Company

League Republic: Revolutionizing Sports League Management

Figure 2- 1 Logo of League Republic

League Republic is a platform which aims at change the management method of sports event. The platform use simplified sports events management, providing a set of tools to increase the efficiency, simplify the process and improve the experience of participants and organizers.

With the development of sports and requirements of management project increasing, League Republic provide a platform to assist the organizers to make match schedules, score tracking, player management and communication with participants. The digital way reduces the paper work and decrease the possibility of mistakes to increase the fairness.

The functionalities of Legue Republic can be used in any sports league. And the platform includes the customize functionalities such as match scheduling, player data collection, real-time score tracking and notification, which helps participants interact with audiences.

Using League Republic, sports organizers can focus on promoting the development of the events, get more people involved into the events. Also, the platform provides the data analytics, to help the organizer analysis the performance and improve the activities planning of future.

In the future, League Republic will expand its functionalities, adding advanced analysis tools and ai-driven predict function to improve the management of sports league. With its user-friendly interface and methods, League Republic becomes an important part in the market.

3. Risk Management

3.1. Risk Identification and Assessment

Risk management begins with identifying potential threats that could hinder the success of a project. For this project, we have identified the following key risks:

- **Technical Risk**: It includes the difficulties in using new technique, the complexity of system and the performance especially in high concurrency and the real-time score tracking. To realize these functionalities may harder than expected and cause extra cost.
- **Operational Risk**: Daily operational risks include data loss, equipment failures, and network issues that could disrupt service availability and user experience.
- **Market Risk:** Changes in consumers' demand, competitor actions, and market trends all bring risks. A decrease in demand or a better solution coming out could impact the project's success.
- **Compliance and Legal Risk**: Privacy protection, data security regulations, and potential legal issues.

3.2. Risk mitigation strategies

To reduce the identified risks, we have developed the following mitigation strategies:

- **Pre-research of New Technology**: Thorough research and testing of the technology employed before to implementation will ensure that it is suitable for the project.

- **Code Review and Testing**: Strict code review and automated testing processes will be implemented to maintain high code quality and system stability and reduce the probability of technical failures.
- **Backup and Disaster Recovery Plan**: Regular data backups and a disaster recovery plan are also in place to prevent data loss and ensure quick recovery in the event of an issue.
- **Monitoring and Alarm System**: Establish a real-time monitoring and alarm mechanism for the system, track the health status of the system, and deal with system abnormalities in a timely manner.
- **Market Research**: Ongoing market research will help to spot changes in market trends and competitor activity, allowing us to quickly perceive changes in consumer demand or competition.
- **Flexible Product Design**: The product will be designed with flexibility in mind, allowing for rapid adaptation of system functionality to meet changing user needs and market changes.
- **Legal Advice**: Consult with legal counsel regularly to ensure compliance with relevant laws and regulations.
- **Data Security Measures**: Strong data security protocols will be implemented to protect user privacy and sensitive information, reducing the risk of data breaches or illegal acts.

4. Requirements Analysis

4.1. User

Players:

Mainly students and amateur football enthusiasts, usually between the ages of 16-30, mainly school teams. Familiar with mobile and web applications related to sports or event management. Players fixtures, team information and individual performance statistics firstly. They rely on notifications for match updates and love the ability to keep track of their team and league match data.

Coach:

Usually professional guiders or amateurs, aged between 20-50 years. Familiar with relevant platforms for team management or communication. Coaches mainly manage and submit team lineups, track player performance, prepare for games, and submit team information. They wanted to streamline the process of communicating with players and provide tools to back it up.

Referee:

Licensed referees or amateur referees, usually appointed to officiate at a particular match or tournament, are between the ages of 20-45. Familiar with basic digital platforms for fixture scheduling and report submission. Referees focus on the clear assignment of match tasks and the tools for submitting match results and referee reports.

Organizer:

The team of event leaders or sports management professionals, usually between the ages of 25-45. Proficiency in the use of software and platforms for the planning and management of sporting events. The organizers focus on arranging the tournaments, managing the tournaments, and ensuring smooth communication between all parties. They needed powerful tools to support logistics such as event registration, referee assignment, and venue management.

Audience/visitor:

Ordinary spectators interested in campus football events, age range 16-60 years old. Familiar with basic web and mobile application operations. The audience mainly focuses on the schedule, the result of the match, and the statistics of the player or team. They want easy access to public information without having to sign up for an account.

4.2. User Story

User Story 1 - The story of office worker Li Ming

- **Role:** Li Ming, a gym teacher at a school, teach students every day and manage the football league in his school
- **Demand:** Due to the busy work, Li Ming wants to find a platform to assist him to manage the football league which can save his time.
- **Story:** As a teacher and football league organizer, Li Ming needs to spend lots of time to manage the match schedule and collect the match results. But he also teaches students in his class so he doesn’t have much time on football league. He needed a system to generate the match schedule automatically and collect the result by digital way.
- **Standard user storytelling:**

As a busy gym teacher and football league organizer (Li Ming), I want a platform that can automatically generate match schedules and digitally collect match results, so that I can save time and efficiently manage the league while focusing on teaching my students.

User Story 2 - The story of student-athlete Zhang Wei

- **Role:** Zhang Wei, a university student who is also a member of the school football team.
- **Demand:** Due to his busy work of studying and training, Zhang Wei wants to access match schedules, team updates, and manage his personal data easily through an online system.
- **Story:** As a student and an athlete, Zhang Wei is busy in his studies and football training. He often misses important announcements or match updates. He needs a system that allows him to stay informed about match schedules, team statistics, and league standings anytime, anywhere, so he can plan his time effectively.
- **Standard user storytelling:**
  As a university student and football player (Zhang Wei), I want to access an online platform to check match schedules and team updates, so that I can manage my time better and stay informed about my team's activities.

4.3. Functional Requirements

- **User Registration and Account Management**: Users (players, coaches, organizers, and referees) must be able to create and manage their accounts in the system. Players can update their personal information, such as their name, contact details, and team affiliation. Coaches and organizers can manage team list and communicate with players.
- **Team and Player Management**: Organizers should be able to create and manage teams, assign players to teams, and update team lists. Players can be added or removed from the team, and the team leader and coaches can update the player status, view and manage team details.
- **Match Scheduling and Management**: Organizers create, schedule, and manage tournaments by setting the tournament period, location, teams, and duration, and the system will automatically generate the schedule. Organizers can update match results, and players and coaches can view upcoming and past match details. The automatic scheduling function avoids overlapping game times and ensures smooth operation.
- **Tournament Bracket Creation and Updates**: The system should dynamically generate and display match results based on the number of teams and the format of the game. The organizer can update the rankings based on the results of the tournament and

automatically adjust the team rankings for the knockout or round-robin round-robins.

- **Live Scoring and Match Updates**: The results of the competition can be updated in real-time during the competition. Scores and match updates should be visible to all users.
- **Player and Team Statistics**: The system must track individual player stats such as goals, assists, and yellow/red cards. Team stats (wins, losses, goals, goals conceded) should be updated automatically after each match.
- **Event Notification and Alerts**: Users should be notified by email when a match is scheduled, changed, or when an important event (e.g., finals, team registration deadline) is approaching. Reminders of upcoming contests, registration deadlines, and other important events should be automatic.
- **Referee and Match Official Management:** Referees and other match officials should be able to register, be assigned to the game, and be updated on their availability. Organizers can assign referees to specific matches and manage their schedules.
- **Payment and Registration Management**: Teams and players should be able to register for matches and pay entry fees through the system. Payment tracking and confirmation should be integrated, and organizers should be able to get financial reports of the competition.
- **Admin Panel for Organizers**: The system must provide an admin interface for the organizer to manage the content associated with the tournament. Support for the management of this real-time entry of information in matches.

4.4. Non-Functional Requirements

- **Performance**: The system should not have significant delays in loading or data processing.
- **Availability**: The system should be available 24/7, with a maximum allowable downtime of 1% per month for maintenance.
- **Reliability**: The system should be reliable, with an uptime of 99.9%, ensuring that match schedules and results are consistently updated.
- **Backup and Recovery**: Data should be regularly backed up and can quickly recover data in the event of a system failure.

5. Research Methodology

5.1. Development Methodology

In the project, I will use Agile Development which is widely used in software development, because of its ability to handle change requirements. It supports short-time iteration and each iteration period is one or two weeks.

5.2. Tools and Technologies

Front-end: React

Create a user-friendly interface with React that supports the integration of UI libraries and frameworks. Enhance code reusability and ensure maintainability through component-based development.

Back-end: Spring Boot

Use Spring Boot as back-end frame, provide a rapid develop environment and support comprehensive security configurations and easy-to-integrate management tools. Its framework supports distributed architectures, making it solve business logic in high efficiency and interact with database and microservice easily. It improves the scalability and stability of system.

Database: MySQL

MySQL is a well-known rational database used in many systems to store structured data. It can support high concurrency condition. It was widely used because of its security and performance.

Cloud Platform: Amazon Web Services (AWS)

AWS is a widely used cloud platform which has security and high availability and provides lost of service.

Testing Framework: Junit & Cypress & Selenium

Junit used for unit testing in Java.
Cypress used for testing the front-end code.
Selenium[4] used for simulator user behavior to test the whole system.

Build Tools: Maven & npm

Maven works as a tool for developing Java application and is good at dependency management. npm is used for front-end project building and dependency management.

Deployment Tool: Docker

Docker is used for containerized application, ensuring consistent developing and producing environment.

Version Control: Git

Git is a widely used tool in software development.

Data Visualization Tools:

Chart.js is suitable for simple competition statistics and performance presentation.

D3.js used to dynamically generate complex interactive data visualizations and present complex statistics.

6. System Architecture

It is very important to consider the needs of hardware and software when designing a system. The architecture of the system should be able to support high concurrency, high availability, and good scalability.

6.1. Hardware Architecture

Service Layer

- **Web Server:** Deal with user request and provide front-end application (React) and API interface (Spring Boot). The server in my project is AWS cloud server.
- **Database Server:** Deal with MySQL database to store all the data such as user info, match and payment records. Periodically back up the database to ensure data recovery capability.

Network Layer

- **Load Balance:** Distribute client requests through a load balancer, such as Nginx or AWS ELB, to ensure traffic balance among multiple Web servers and improve system availability and scalability.

Safety and protection

- **transmission encryption:** All transmitted data in the system is encrypted using the

TLS/SSL encryption protocol, ensuring the security of data transmission between the client and the server. All users' sensitive information should be encrypted in transit using the HTTPS protocol.[5]

- **Identity authentication and permission control:** Use OAuth 2.0 and JSON Web Tokens (JWT) to manage user authentication. OAuth2 is an authorization protocol that generates an authorization token when a user logs in, which can be used as a credential in subsequent requests. JWTs can effectively pass authentication information in a headless architecture.[6]

6.2. Software Architecture

In my project, I choose to use Microservice Architecture which divide the system into several service. Each service is responsible for a single functional module. In my project, it is divided into user management service, match management service, team management service, finance service. The communication between each service realized by RESTful API.[7]

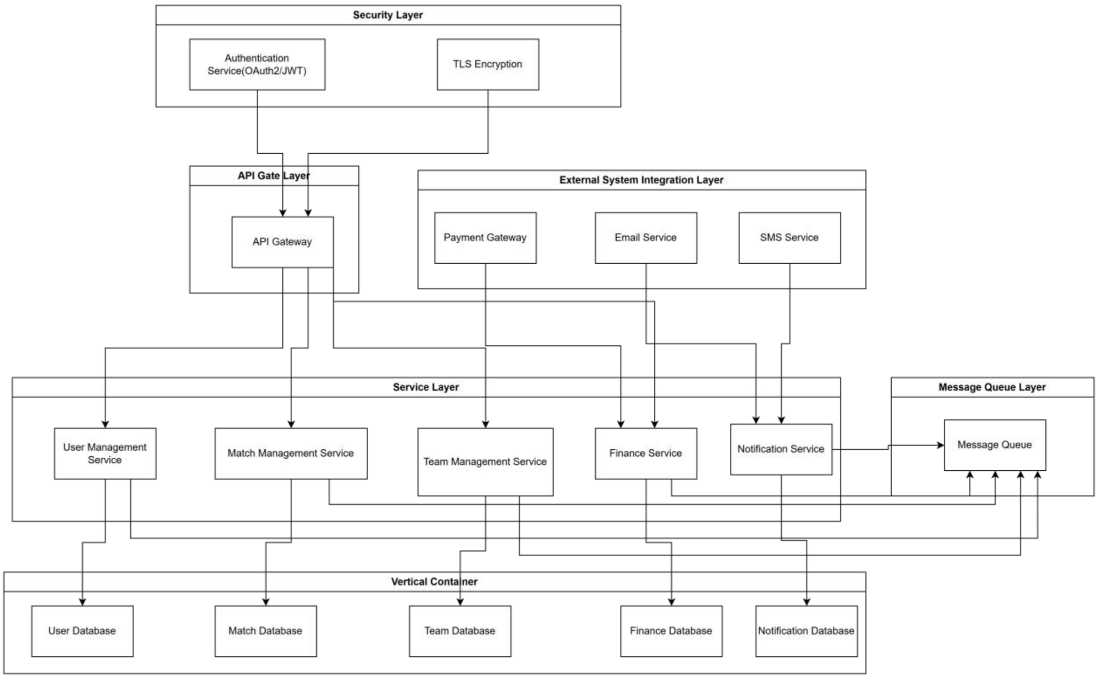


Figure 6- 1 System Architecture Diagram

7. System Design

7.1. Sub-system Definition

This application is divided into four main subsystems: a User sub-system to manage user accounts, a Match sub-system to manage match details, a Team sub-system for team management, and a Finance sub-system for financial management.

7.2. Use Case Diagrams

7.2.1. User Sub-system

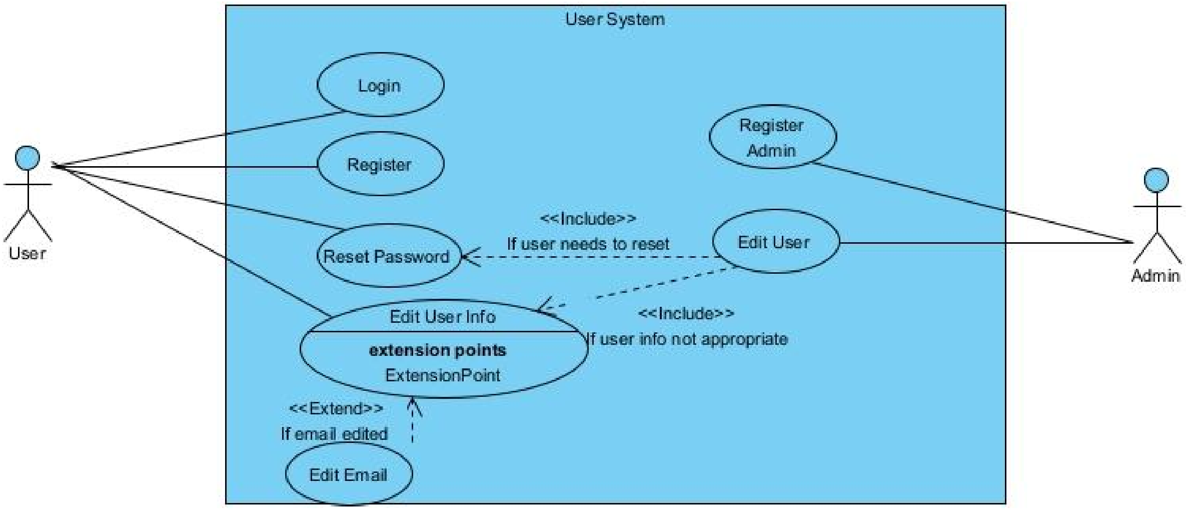


Figure 7- 1 User System Use Case Diagram

7.2.2. Match Sub-system

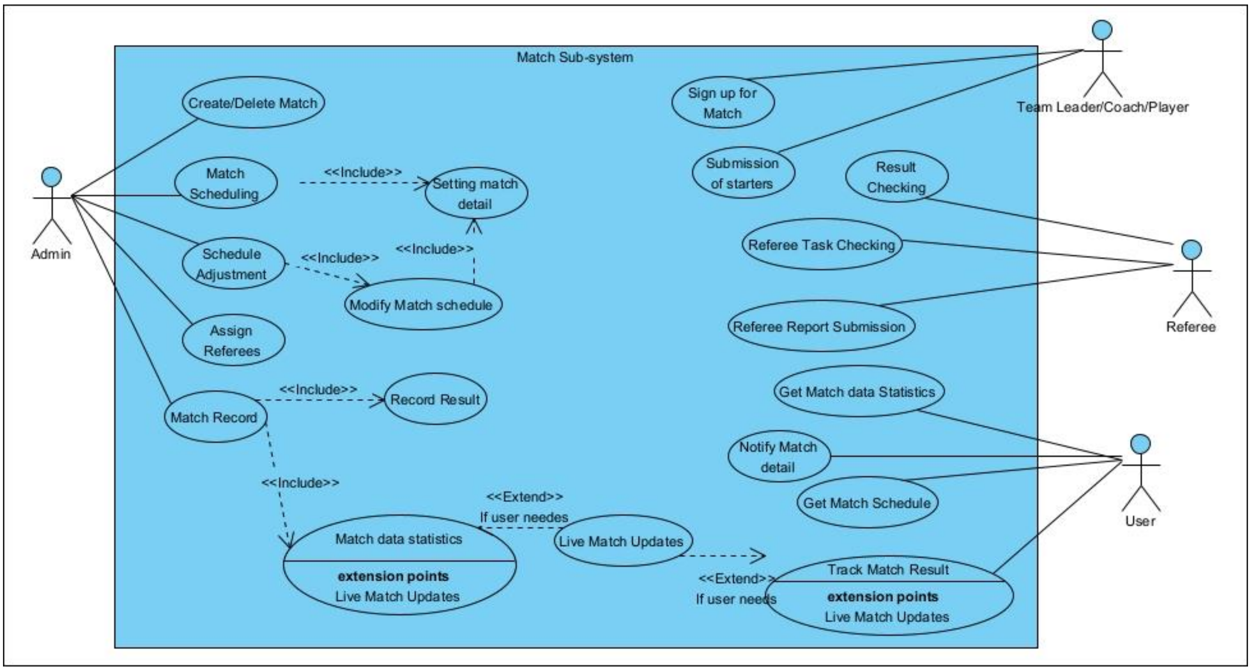


Figure 7- 2 Match System Use Case Diagram

7.2.3. Team Sub-system

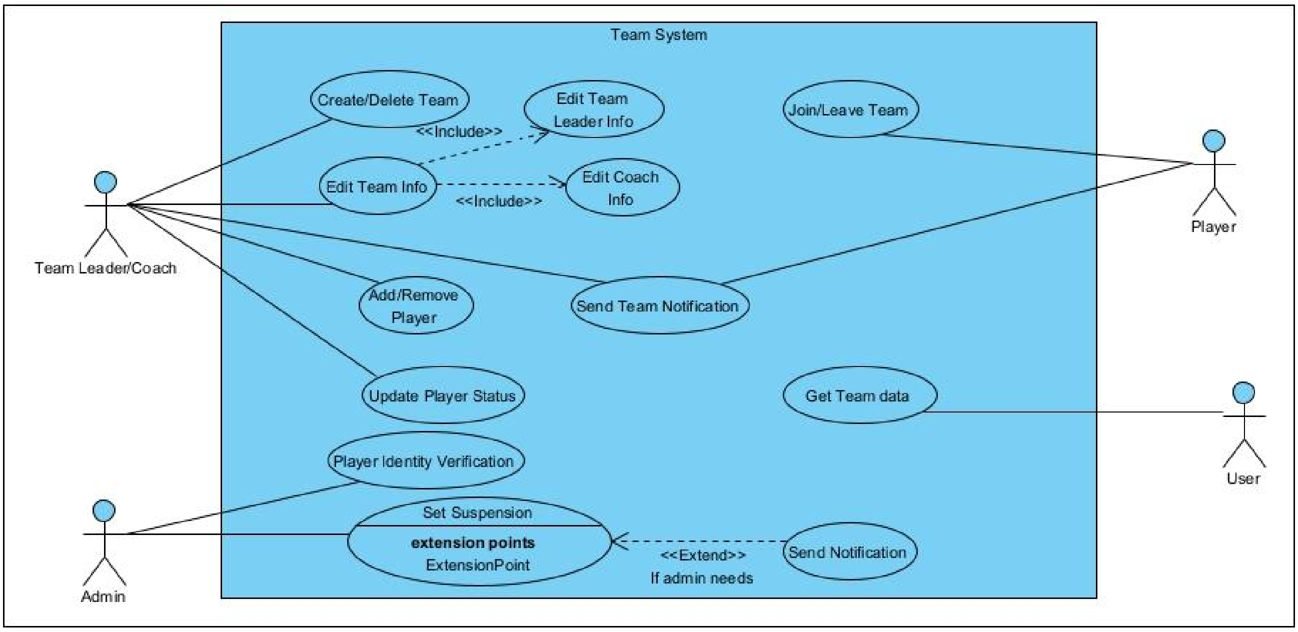


Figure 7- 3 Team System Use Case Diagram

7.2.4. Finance Sub-system

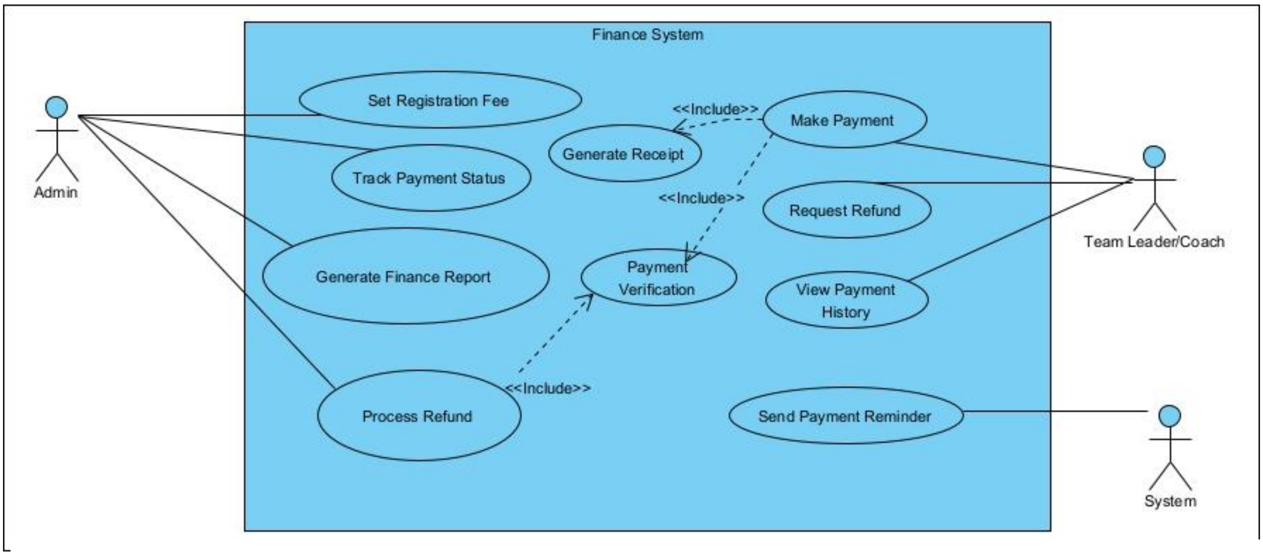


Figure 7- 4 Finance System Use Case Diagram

7.3. Class Diagram

User Sub-system

The class diagram of User subsystem is shown in Figure 6-5. User class can be divided into admin, referee, player, coach and common user, which have different permissions

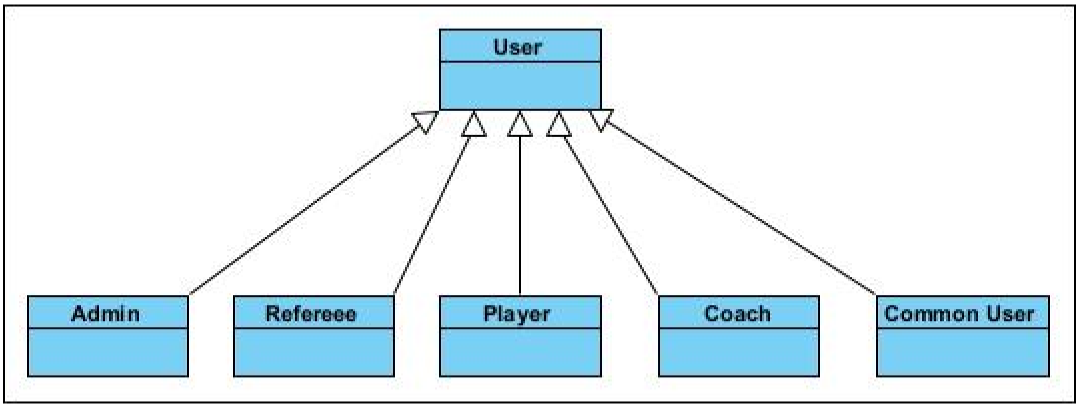


Figure 7- 5 User System Class Diagram

Match Sub-system

The class diagram of Match Sub-system is shown in Figure 6-6. Each tournament has a match schedule. Each schedule contains many matches. Each match has referees and the match result. Each match result consists of match event and notification. And each referee has a referee report.

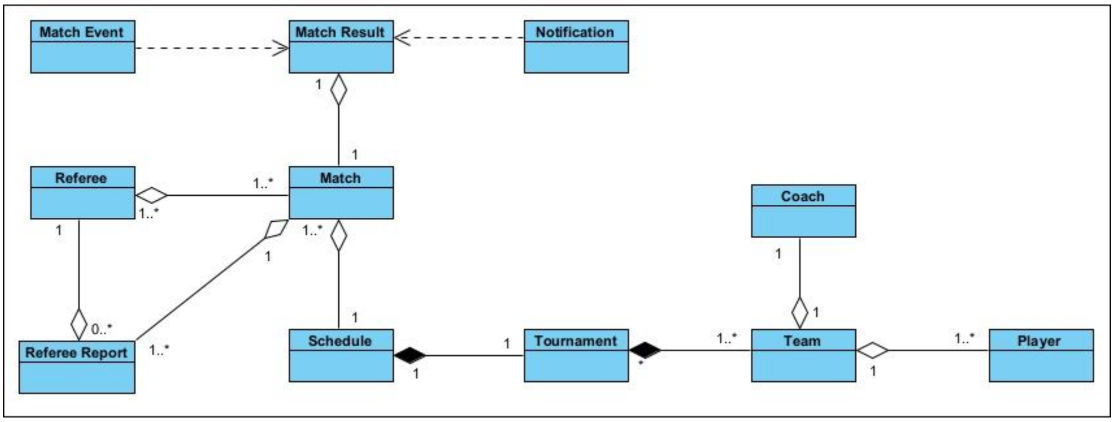


Figure 7- 6 Match System Class Diagram

Team Sub-system

The class diagram of Team Sub-system is shown in Figure 6-6. Each team consists of a couch and many players. Each Team joins in tournament or not. Each player contains a player profile and match data.

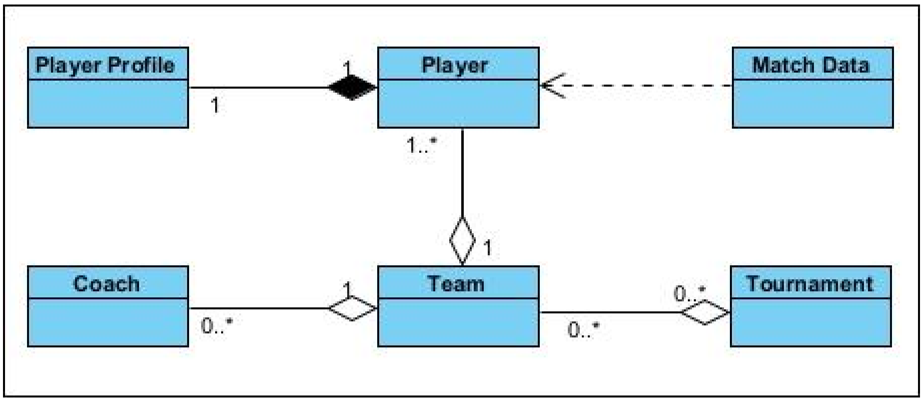


Figure 7- 7 Team System Class Diagram

Finance Sub-system

Each team has a tournament registration fee which is related to financial account. Financial account has payment and transaction. Each payment has a receipt.

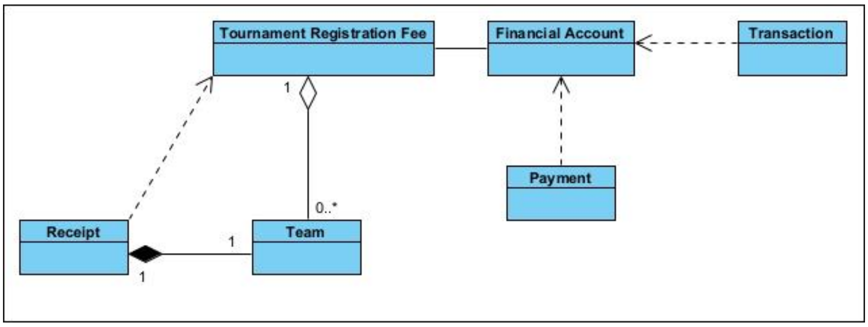


Figure 7- 8 Finance System Class Diagram

7.4. Sequence Diagram

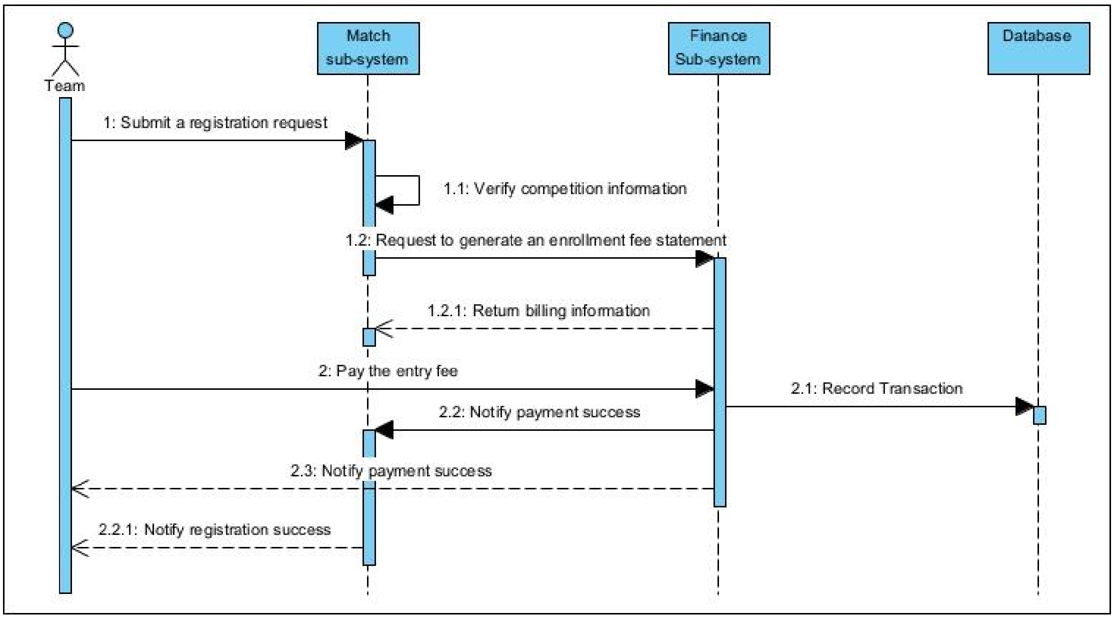

Team
Match sub-system
Finance Sub-system
Database
1: Submit a registration request
1.1: Verify competition information
1.2: Request to generate an enrollment fee statement
1.2.1: Return billing information
2: Pay the entry fee
2.1: Record Transaction
2.2: Notify payment success
2.3: Notify payment success
2.2.1: Notify registration success

Figure 7- 9 Competition registration process

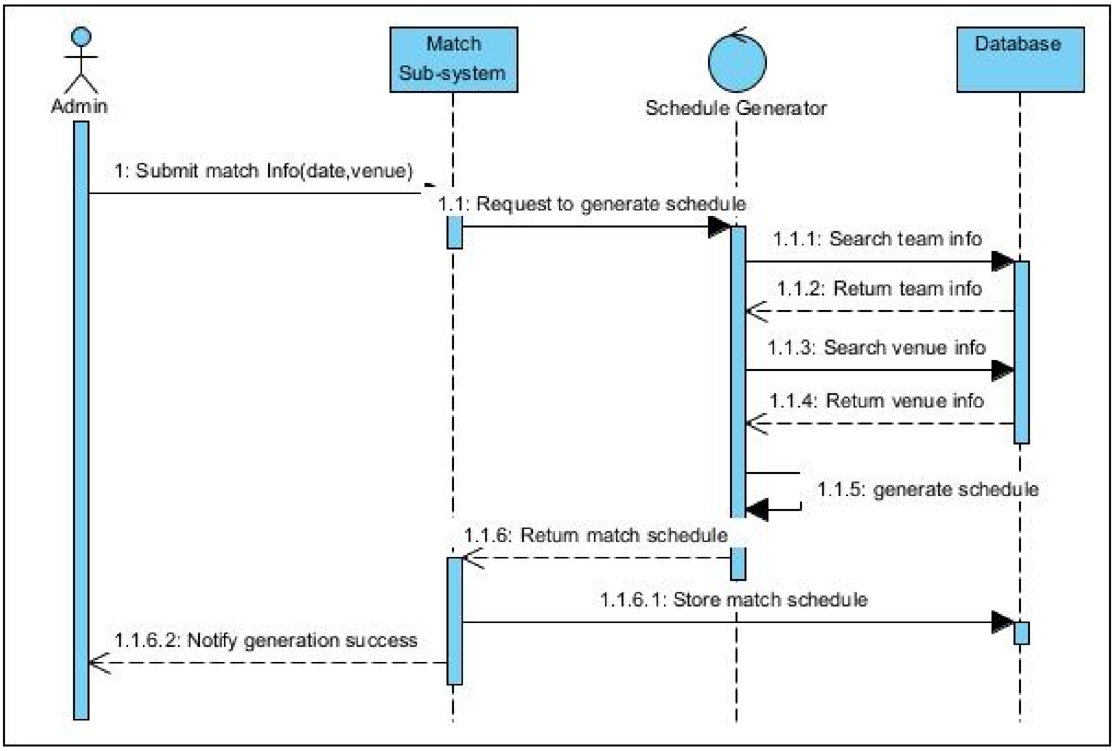


Figure 7- 10 Schedule Generating Process

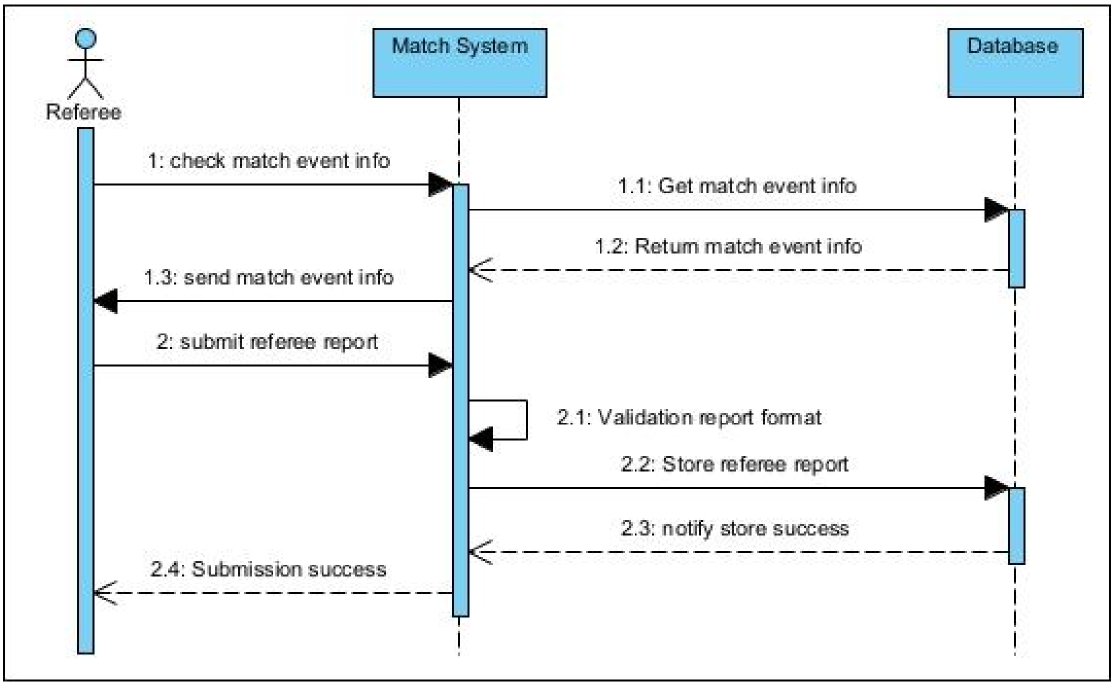

Referee
Match System
Database
1: check match event info
1.1: Get match event info
1.2: Return match event info
1.3: send match event info
2: submit referee report
2.1: Validation report format
2.2: Store referee report
2.3: notify store success
2.4: Submission success

Figure 7- 11 Match Event Checking and Referee Report Submission Process

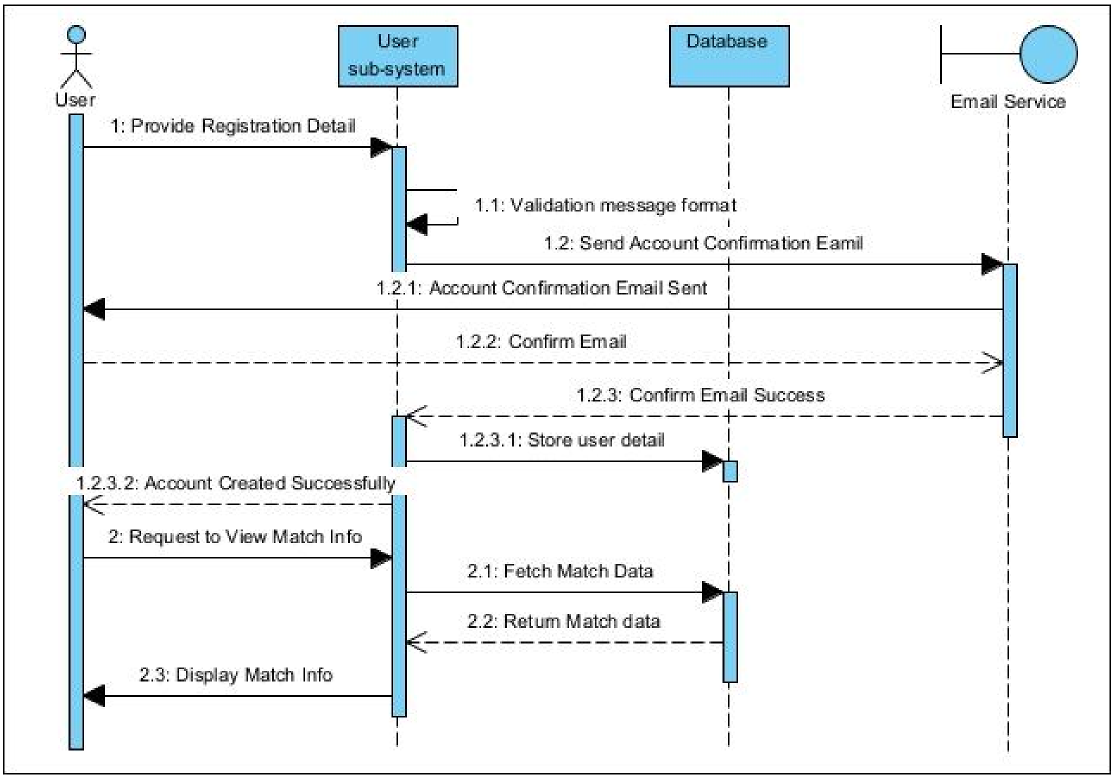

User
User sub-system
Database
Email Service
1: Provide Registration Detail
1.1: Validation message format
1.2: Send Account Confirmation Eamil
1.2.1: Account Confirmation Email Sent
1.2.2: Confirm Email
1.2.3: Confirm Email Success
1.2.3.1: Store user detail
1.2.3.2: Account Created Successfully
2: Request to View Match Info
2.1: Fetch Match Data
2.2: Return Match data
2.3: Display Match Info

Figure 7- 12 User Registration Process

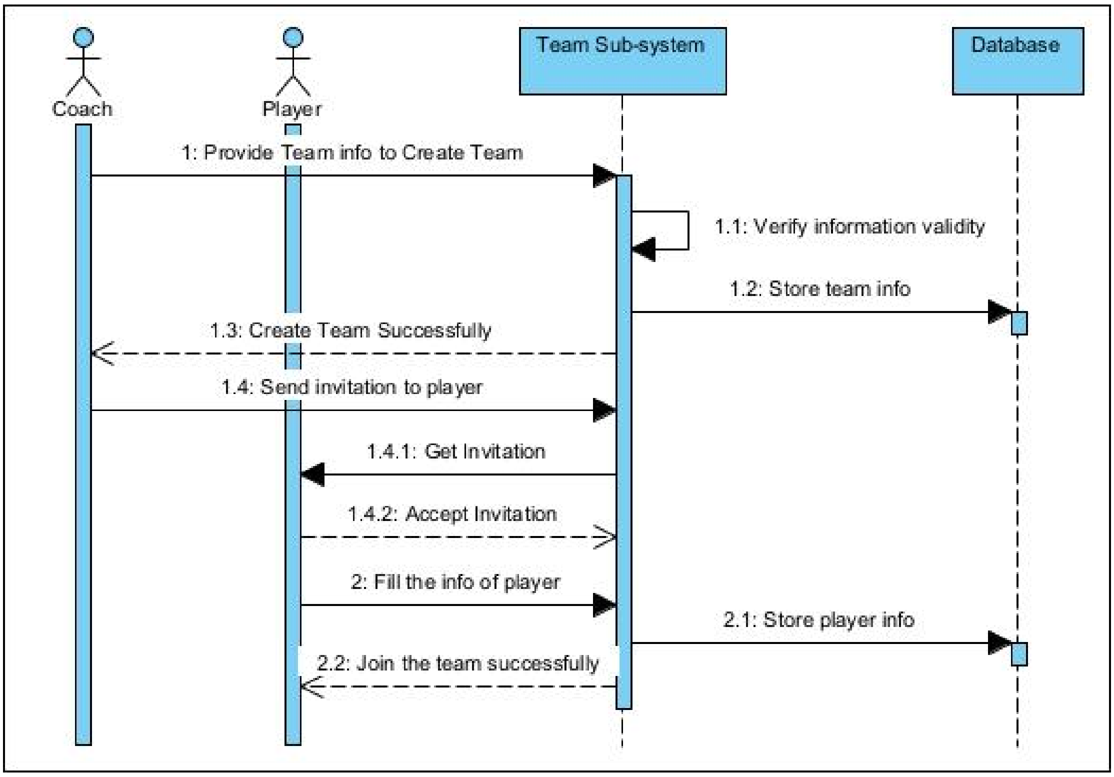

Coach
Player
Team Sub-system
Database
1: Provide Team info to Create Team
1.1: Verify information validity
1.2: Store team info
1.3: Create Team Successfully
1.4: Send invitation to player
1.4.1: Get Invitation
1.4.2: Accept Invitation
2: Fill the info of player
2.1: Store player info
2.2: Join the team successfully

Figure 7- 13 Team Creation and Management Process

7.5. Entity-Relationship Diagram

The project use MySQL, a kind of rational database. It is important to design a well-structured database to manage the data in my project. To realize the database design visualization, I drew an Entity-Relation diagram showed in Figure 6-14 which shows the entities, attributes, and relationships in the database.

In this chart, a key in the front of the line means primary key and each attribute has a primary key. And the italic means it is a foreign key. When there is a many-to-many relationship between two entities I added an intermediate table to decrease the complexity and improve the extendibility of the database.

Here the entity-relationship diagram only shows the relationship of each table in the database. I will divide the data table into five database based on the microservice architecture including user database, team database, match database, finance database, and notification database to ensure each service has an independent database.

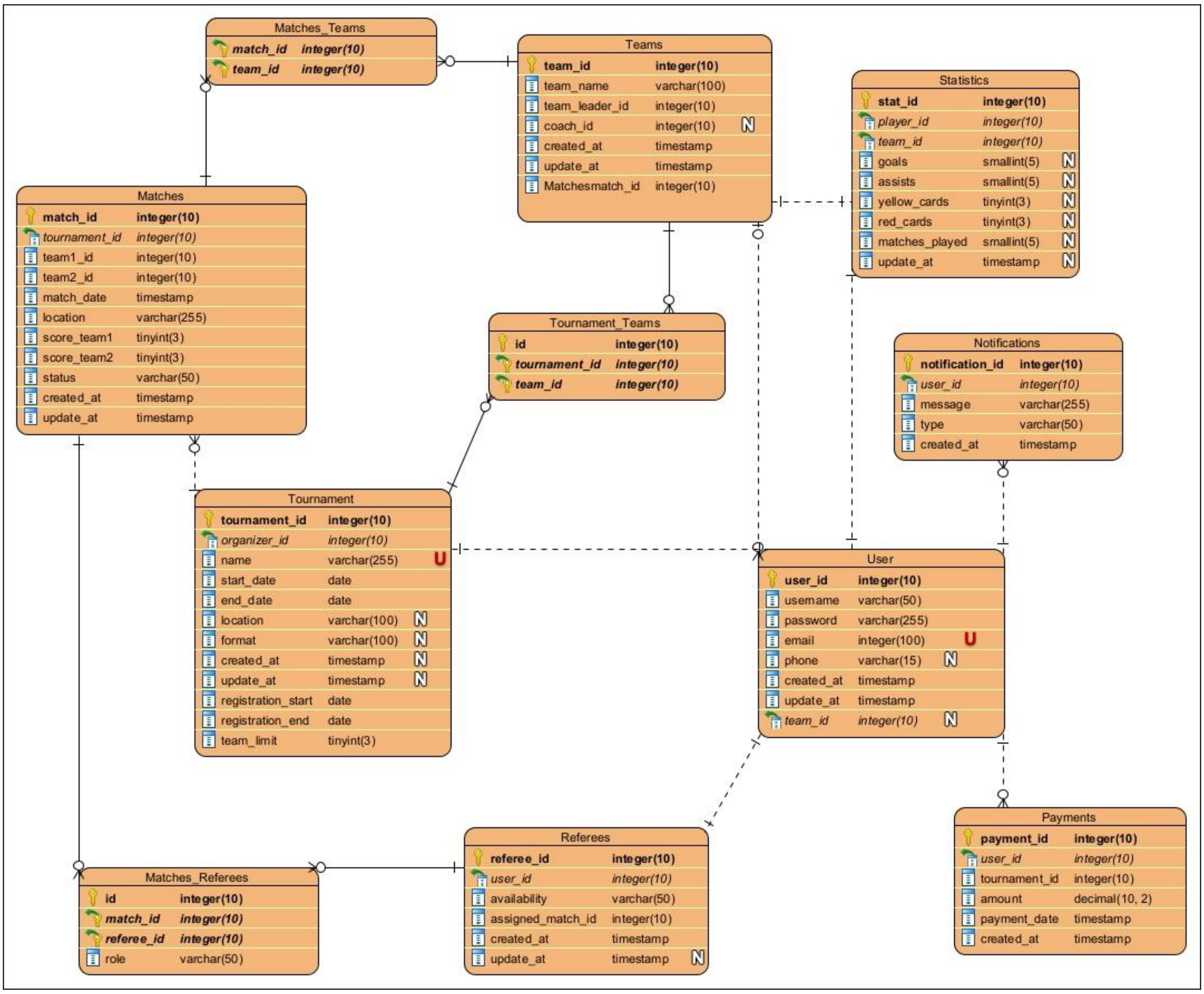
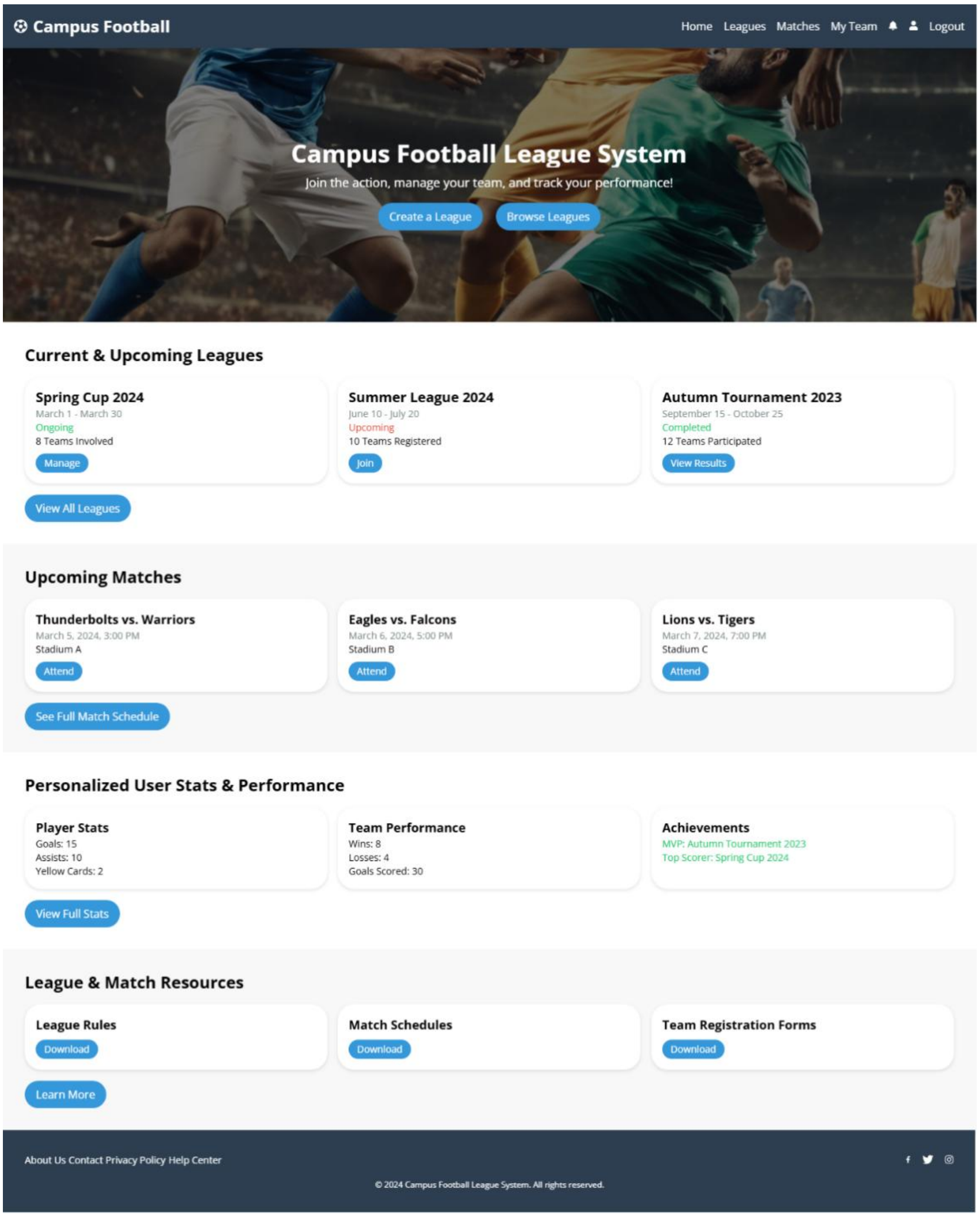


Figure 6- 7- 144 Entity-Relation Diagram

8. Visual Design

Home Page



Campus Football League System

Join the action, manage your team, and track your performance!

Current & Upcoming Leagues

Spring Cup 2024
March 1 - March 30
Ongoing
8 Teams Involved

Summer League 2024
June 10 - July 20
Upcoming
10 Teams Registered

Autumn Tournament 2023
September 15 - October 25
Completed
12 Teams Participated

Upcoming Matches

Thunderbolts vs. Warriors
March 5, 2024, 3:00 PM
Stadium A

Eagles vs. Falcons
March 6, 2024, 5:00 PM
Stadium B

Lions vs. Tigers
March 7, 2024, 7:00 PM
Stadium C

Personalized User Stats & Performance

Player Stats
Goals: 15
Assists: 10
Yellow Cards: 2

Team Performance
Wins: 8
Losses: 4
Goals Scored: 30

Achievements
MVP: Autumn Tournament 2023
Top Scorer: Spring Cup 2024

League & Match Resources

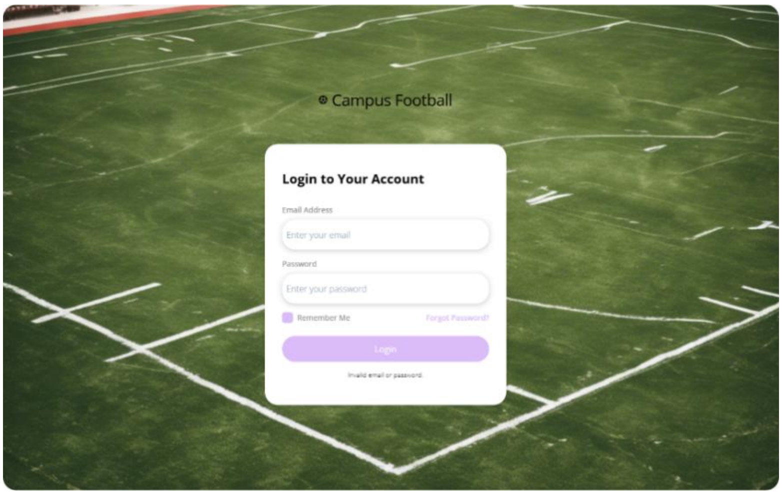
League Rules

Match Schedules

Team Registration Forms

© 2024 Campus Football League System. All rights reserved.

Figure 8- 1 Home Page



⚽ Campus Football

Login to Your Account

Email Address

Enter your email

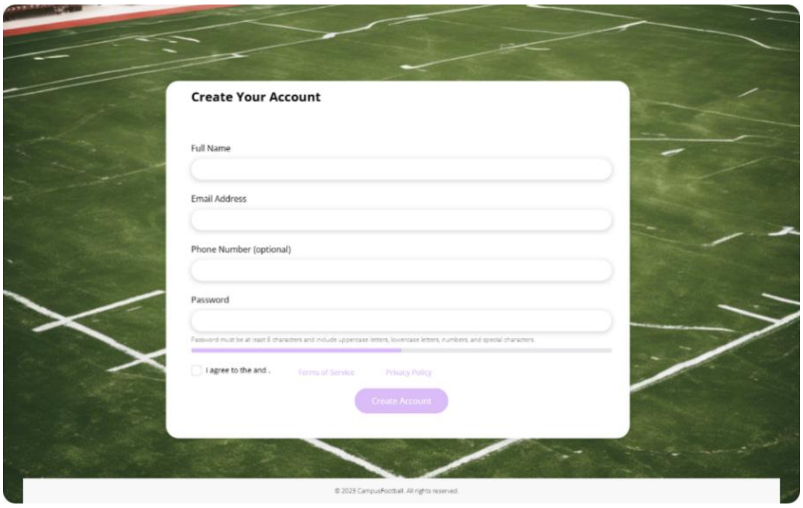
Password

Enter your password

☑ Remember Me

Invalid email or password.

Figure 7- 1 Login Page



Create Your Account

Full Name

Email Address

Phone Number (optional)

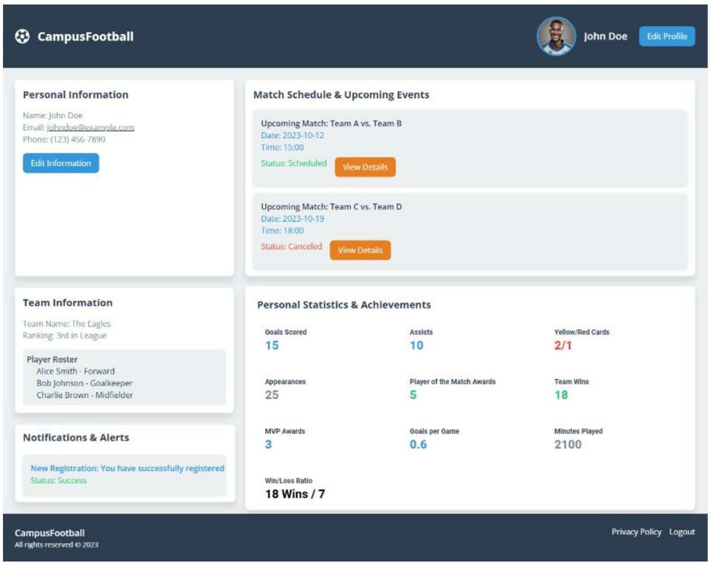
Password

Password must be at least 8 characters and include uppercase letters, lowercase letters, numbers, and special characters.

□ I agree to the and .

© 2023 CampusFootball. All rights reserved.

Figure 8- 2 Register Page



⚽ CampusFootball

John Doe

Personal Information

Name: John Doe
Email: johndoe@example.com
Phone: (123) 456-7890

Match Schedule & Upcoming Events

Upcoming Match: Team A vs. Team B
Date: 2023-10-12
Time: 15:00
Status: Scheduled

Upcoming Match: Team C vs. Team D
Date: 2023-10-19
Time: 18:00
Status: Canceled

Team Information

Team Name: The Eagles
Ranking: 3rd in League

Player Roster
Alice Smith - Forward
Bob Johnson - Goalkeeper
Charlie Brown - Midfielder

Notifications & Alerts

New Registration: You have successfully registered
Status: Success

Personal Statistics & Achievements

Goals Scored 15	Assists 10	Yellow/Red Cards 2/1
Appearances 25	Player of the Match Awards 5	Team Wins 18
MVP Awards 3	Goals per Game 0.6	Minutes Played 2100
Win/Loss Ratio 18 Wins / 7		

CampusFootball
All rights reserved © 2023

Figure 8- 3 Personal User Info Page

9. Schedule

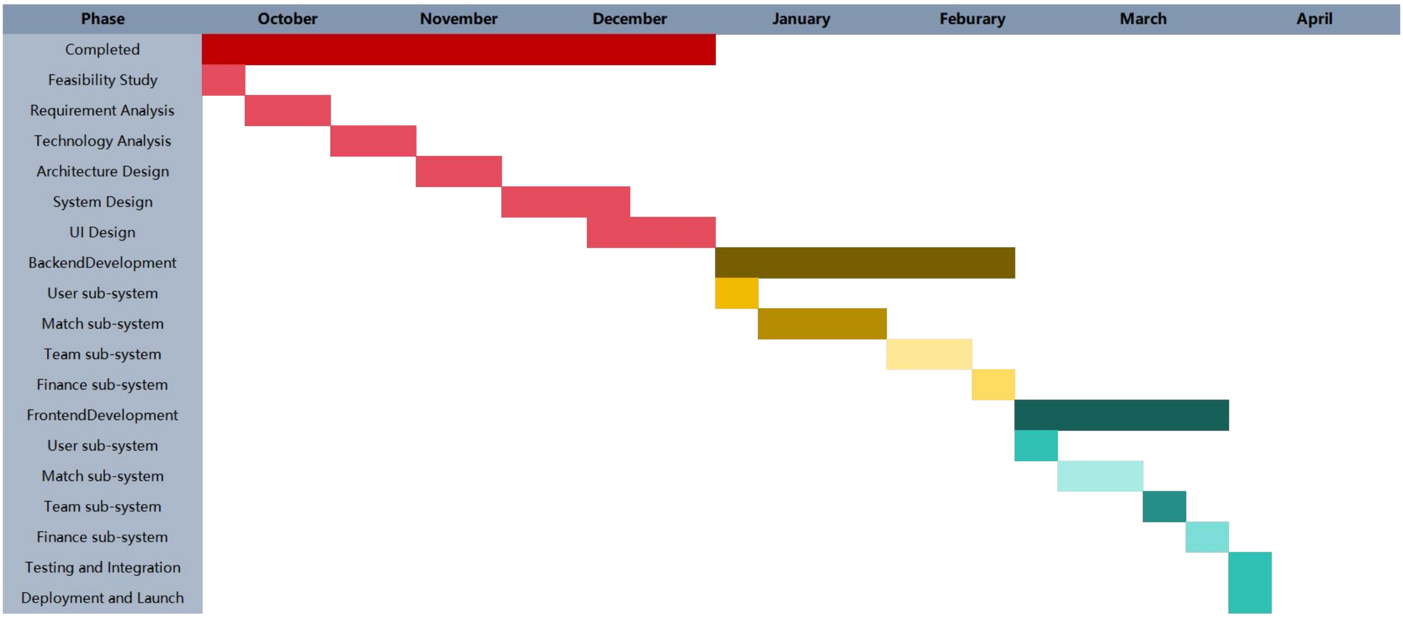


Figure 9- 1 Gantt Chart

10. Perference

[1] Chen, S., & Zhang, X. (2020). The Development and Management of Campus Sports Activities: Challenges and Opportunities. Journal of Physical Education and Sports Science, 12(3), 45–57.

[2] European School Sports Day (2022). European School Sports Day Report.

[3] Zhang, Y., & Li, H. (2021). Adopting Digital Solutions for Sports Management: A Case Study on Campus Sports. Technology in Sports Review, 9(4), 101–116.

[4] SeleniumHQ. (n.d.). Selenium documentation. [Online]. From https:// www. selenium. dev/ documentation/

[5] IETF. (2000). RFC 2818: HTTP Over TLS. Internet Engineering Task Force. https://tools.ietf.org/html/rfc2818

[6] de Almeida, P. (2016). OAuth 2.0 and JWT: Securing the API Ecosystem. *O'Reilly Media*. ISBN 978-1-4919-2321-0.

[7] Wang, X. (2022). *Microservices Architecture Literature Review and Case Studies*. Zhihu. Retrieved from https://zhuanlan.zhihu.com