Software Requirement Specification for Sports Club Management System

Table of contents:

1. Introduction

– 1.1 Purpose

– 1.2 Scope

– 1.3 Definitions. Acronyms, and Abbreviations

– 1.4 References

– 1.5 Overview

2. Overall Description

– 2.1 Product Perspective

– 2.2 Product Functions

– 2.3 User Characteristics

– 2.4 Constraints

– 2.5 Assumptions and Dependencies

3. Specific Requirements

– 3.1 External Interfaces

– 3.2 Functions

– 3.3 Performance Requirements

– 3.4 Logical Database Requirements

– 3.5 Design Constraints

– 3.6 Software System Quality Attributes

–3.7 Object Oriented Models

4. Appendices

5. Index

---

1. Introduction:

1.1 Purpose:

The purpose of this document is to specify the requirements for a sports club management system that supports user registration, membership management, event scheduling, fee handling, and notifications.

1.2 Scope:

This system will automate member registration, manage club events, track payments, and send notifications.

1.3 Definitions. Acronyms, and Abbreviations:

1.3.1. Definitions:

Admin: A user who has administrative privileges to manage system users, schedules, fees, and other essential functions. Responsible for overseeing and controlling the club's operations within the system.

Member: A registered individual who is part of the sports club and has access to member-related features, such as viewing their profile, schedules, and managing payments.

Staff: Individuals responsible for handling specific club activities, including managing events, member communication, and other operational tasks.

Visitor: A non-registered user who can browse the website, but cannot access full member or admin functionalities until they register or log in.

Payment Gateway: An external system integrated into the club's software for processing payments, such as for membership fees or event-related charges.

Notification System: The feature within the system that sends updates, reminders, or alerts to members, staff, or admins (e.g., match notifications or fee due alerts).

User Profile: The personal information page for each user (member, staff, admin) where they can view and update their personal and contact details.

Reporting: The generation of reports on system activities like member registrations, payments, event participation, etc.

1.3.2. Acronyms:

SRS – Software Requirements Specification
UI – User Interface
API – Application Programming Interface (for integration with external systems like payment gateways or email services)
CRUD – Create, Read, Update, delete (refers to basic database operations)
DBMS – Database Management System (software used to manage and organize the database)
SCMS_ Sports Club Management System

1.3.3. Abbreviations:

VIP – Very Important Person (can refer to a special member or a distinguished guest in the system)
SMTP – Simple Mail Transfer Protocol (used for sending email notifications).

1.4 References

IEEE 830-1998 Standard, 'Software Requirements Specification'.

1.5 Overview:

The **Sports Club Management System** (SCMS) is designed to streamline operations of a sports club by providing an integrated solution for managing members, events, payments, and notifications. The system offers a user-friendly interface for different user roles, including **Admins**, **Staff**, and **Members**. Admins will have full control over the system, such as managing users, setting up events, and generating reports. Staff will be responsible for scheduling events and communicating with members. Members can register, view their schedules, and make payments.

The system will interface with external services like payment gateways (e.g., PayPal) for membership fee processing and email services (e.g., SendGrid) for sending notifications. Performance-wise, the system will support up to **500 concurrent users** and ensure quick response times. It will also support easy scalability as the club grows.

Overall, the SCMS will offer a seamless and efficient solution to handle the core operations of the sports club, ensuring a smooth experience for users at all levels.

2. Overall Description:

2.1 Product Perspective:

This system will integrate with an external payment gateway for processing fees.

2.2 Product Functions:

The **Sports Club Management System** will offer the following key functions:

2.2.1. Manage Members:

Admins can manage member profiles, including adding new members, updating existing records, and deleting memberships. Members can also update their personal information and track their payment status.

2.2.2. Schedule Matches:

The system will allow the scheduling of matches, training sessions, and events. Admins and staff can create, edit, or delete events and notify participants about important details and changes.

2.2.3. Manage Fees:

Admins will be able to define membership fees, track payments, and process transactions using integrated payment gateways. Members can view and pay their dues online.

2.2.4. Send Notifications:

The system will send notifications to users regarding upcoming matches, events, membership fee reminders, and other important updates. Notifications will be sent via email and SMS.

2.2.5. View Personal Profile:

Members can access and update their personal profiles, view match schedules, check payment history, and update their contact information.

2.2.6. Generate Reports:

The system will generate various reports for admins and staff, including attendance, financial transactions, and match results. Reports can be exported as CSV files.

2.2.7. Authenticate Users:

The system will authenticate users based on roles (Admin, Staff, Member, Visitor) to ensure access control and privacy. Only authorized users will be able to access certain functionalities.

2.2.8. Search and Filter Records:

Admins and staff can search for specific members, events, or financial records and filter them based on different criteria, such as name, date, or payment status.

2.2.9. Manage Staff:

Admins will be able to manage staff roles and responsibilities, assign tasks, and monitor staff activities in the system.

2.3 User Characteristics:

Admins and Coach should have a technical background, while members should be able to use the system without technical skills.

2.4 Constraints:

The system should be compatible with modern web browsers, and should be built in python programming language.

2.5 Assumptions and Dependencies:

- The system assumes that users have proper access to the internet.
- The system assumes that all users will have **unique usernames** and **passwords** to ensure security.
- The system assumes that users will use **modern web browsers** (such as Chrome, Firefox, Safari, or Edge) to access the platform.
- The system depends on the availability and proper integration of **third-party payment**
- **gateways for** processing member fees (e.g., PayPal, Stripe).
- The system is dependent on a **web hosting provider** to ensure that the platform is available online and performs reliably.

3. Specific Requirements:

3.1 External Interfaces:

The system will integrate with PayPal and Jazz cash for processing membership fees.

3.2 Functions:

The admin can add a new member by entering their personal details and membership plan.

3.3 Performance Requirements:

The system should be able to handle up to 500 concurrent users having response time should be less than 2 seconds and user interface should be robust and aesthetic.

3.4 Logical Database Requirements:

The system should store member data, match schedules, and payment records in a relational database.

3.5 Design Constraints:

The system must be web-based, developed in python programming language and compatible with the latest version of Chrome, Firefox, and Safari etc.

3.6 Software System Quality Attributes:

Reliability: The system should be highly reliable with minimal downtime.

Scalability: The system should be able to scale as more members and events are added.

Security: The system should encrypt sensitive user data and require strong authentication for admin functions.

3.7 Object Oriented Models:

Use Case Diagram:

The **Use Case Diagram provides** a high-level view of the system's functionality and shows how different actors (Admin, Member, Staff, Visitor) interact with the system. This diagram is essential for understanding the major actions users can perform in the system, such as **Managing Members** or **Scheduling Matches**.

Class Diagram:

The **Class Diagram defines** the system's core classes and their attributes and methods. This diagram illustrates how entities like **Member**, **Staff**, **Payment**, and **Match** are structured and related. For instance, the **Member** class includes attributes like `name`, `email`, and methods like `register ()`.

Sequence Diagram:

The **Sequence Diagram** demonstrates the interaction flow between different objects during the **Member Registration** process. This diagram highlights the sequence of messages exchanged between the **Member**, **Admin**, and **Database** during the registration flow.

4. Appendices

4.1 Database Schema:

Members Table: Member ID, Name, Membership Type
Payments Table: Payment ID, Member ID, Amount
Matches Table: Match ID, Date, Teams.

4.2 Glossary of Terms:

Admin: The user who manages the club's activities.
Member: A person registered with the club.
Payment System: The external system to process membership fees.

4.3 External System Integration:

a) Payment Gateway: You plan to use an external payment system (e.g., PayPal) to process payments.

b) Email Notifications: The system will notify users about events or payment reminders.

5. Index:

Introduction..... Page 2
Admin .....Page 2
Members .....Page 2
Overall Description ..... Page 4
Manage Members ..... Page 4
Schedule Matches ..... Page 5
Send Notifications ..... Page 5
Notifications ..... Page 5
Specific Requirements ..... Page 6
Payment .....Page 6