Software Requirements Specification

for

Sports Club Management System

Version 1.0 approved

Prepared by Nour Abdelmoneim, Michael Kwan, Arb Mehmeti, Nahom Molla

New York University Abu Dhabi

April 12th, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product is a sports club management system release number 1. The scope of this SRS will cover the entire system that will be used to manage all the analytics and data for sports teams.

1.2 Document Conventions

This SRS follows the template provided by IEEE. Priorities for higher-level requirements are assumed to be inherited by detailed requirements.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for developers, project managers, marketing staff, and documentation writers. This SRS contains information about the functions, classes, interfaces, and features of the Sports Club Management System that allow it to satisfy all of its requirements. Readers should start at the overall description of the system followed by the features to understand what the system is meant to do. Then, the reader can go through the external interface requirements and other nonfunctional requirements as necessary. Developers and project managers will have to read the entire SRS to get all the details that they will need to complete the project.

1.4 Product Scope

The sports club management system is a system that will store data and analytics for a sports club. The benefits is that the system will be able to store and access more information than the previous paper-based version. The objective of this system is to help clubs be more organized and have better ease of access of data, especially if the club has a lot of data to keep track of. The goal is for anyone to be able to manage a sports club because this system will make that responsibility significantly easier.

1.5 References

This SRS references the srs_template-IEEE template and the System Request document.

2. Overall Description

2.1 Product Perspective

This sports club management system is an online system that would replace the existing paper based system.

2.2 Product Functions

1. **Scheduling functions**
    a. View personal schedules (coaches/players)
    b. View competition schedules
    c. Edit competition schedules
    d. Edit personal schedules

2. **Registration functions**
    a. Register for new class
    b. Cancel existing class registration
    c. Register for competition
    d. Register for membership as new user

3. **Payment functions**
    a. Make payment
    b. Send payment request
    c. Update payment status
    d. Calculate amount owed by player

4. **Stats functions**
    a. Display team stats

5. **Equipment functions**
    a. View available equipment
    b. Return equipment
    c. Checkout equipment
    d. Update equipment status

6. **Other functions**
    a. Display financial report

2.3 User Classes and Characteristics

1. **Players**
    a. Subset of product functions used:
        i. Register for classes
        ii. View coaches' available time
        iii. View personal schedule
        iv. Cancel/ edit class registration
        v. Make payment
        vi. View competition schedule
        vii. View/checkout equipment
        viii. Register for membership as new user

b. Security/ Privilege level: Low privilege
   i. Limited access to information such as payment
   ii. Specificities of coach schedules hidden
   iii. No access to other player schedules
   iv. No access to team/player stats

2. Coaches

a. Subset of product functions used
   i. View/ edit personal schedule
   ii. View player stats
   iii. View team stats
   iv. View competition schedule/ participation
   v. View/checkout equipment

b. Security/ Privilege level: Medium
   i. No access to payment information
   ii. No access to specific player schedule
   iii. Access to team/player stats

3. Administration

a. Subset of product functions used
   i. Manage/view payment status
   ii. Manage registration requests
   iii. Update competition schedule
   iv. View competition schedule/ participation
   v. Update equipment list/status

b. Security/ Privilege level: High
   i. Access to payment information
   ii. Access to all coach/ player schedules

2.4 Operating Environment

This system is a web-based system that should be compatible with any operating system.

2.5 Design and Implementation Constraints

Having a stable internet connection is a constraint for the system. Considering that the registration is going to be online then internet connection is required to successfully complete the registration process. Internet connection is also crucial in backing up the data to the server and the cloud storage.
The system might be constrained by the size of its database which may lead to registrations being put on hold and extend the time it takes to successfully complete the registration.

Another constrain is the access the customers are going to have to the players data. That depends on the data given to the sports club and what the management decides to make public.

2.6 User Documentation

On-line help will be provided should any issues arise while using the system.

2.7 Assumptions and Dependencies

One assumption for the system is that users and customers have the necessary hardware (mobile phone, laptop or tablet) in order to access the system. Considering that the system needs a very low performance hardware whether any of the hardware can run it will not be a problem.
Another assumption is that all the data for the registration does not need to be distributed so there will be only one database where this data is stored.
Another assumption is that each user will have different login credentials and will have to log in everytime for security purposes therefore an option to stay logged in will not be available. An automatic logout feature after a certain period of inactivity will be implemented

3. External Interface Requirements

3.1 User Interfaces

The users of the system will see the login screen when they open the system. They will need to login every time they access the system. The management and the workers will have to select their roll from a drop down menu in order to have access to the information that they have permission to view or edit.

On the next screen the user will have different options on the data they can view and edit. The level of access that the users have is determined by the drop down menu choice (management, finance, coaches, front desk etc.) . The user can make a choice from the available tabs they wish to view/edit.

Depending on the user's choice the next screen will be the selected tab with a search bar where the user can look for a certain parameter they want to view/edit. A pencil sign should show next to each parameter if the user is allowed to make edits. If they choose to make changes a pop up will show asking the user if they wish to continue with the change.

The customers that wish to make a registration will be shown a login screen and they are given an option to make a new registration.

If the user has already registered the next screen will show the status of the registration and the option to upload more files needed for the registration.

If the user is registering for the first time they will have to fill in the required information and submit or save their application.

3.2 Hardware Interfaces

Since it's a very simple system, there are no specific hardware requirements. The system can run on any computer, mobile or tablet. The database server is going to run in the main computer which is going to need a fair amount of storage space and the connection is going to be carried by the operating system on the respective hardware.

3.3 Software Interfaces

The system is a web app that works with any web browsing software. The tools and libraries used are MySQL (database), Python3, Flask and HTML.

3.4 Communications Interfaces

The system utilizes a web browser in HTTP format to communicate with the database.

4. System Features

4.1 Register for Class

4.1.1 Description and Priority

The register for class function should allow players to register for a new class through the classes page on his/ her personal account by picking a coach's name and time slot. Within this function, a player should be able to view a coach's available time slot, then pick one that fits his/her schedule. This function should update both the players' and coaches' schedules accordingly. It should also send an error message if the chosen time slot clashes with the player's schedule. Since registering for classes is one of the most important functionalities for a sports club, this function is of high priority.

4.1.2 Stimulus/Response Sequences

To stimulate the behavior designed for this feature, the user (a player) would have to click on a "register new class" option, which would take him/her to a menu/ list of coaches from which he should choose from. Then, upon choosing a coach, the corresponding schedule will show up and the player can proceed to book the class.

4.1.3 Functional Requirements

REQ-1: In the case that a player chooses a time slot that causes a clash within his/her schedule, the system should produce an error message and not proceed with the booking.

REQ-2: In order for this feature to function, the software must have a database of all coaches' and players' schedules stored.

REQ-3: A fast mechanism that updates schedules in real-time is necessary for this feature to function correctly and efficiently.

4.2 Register for Competition

4.2.1 Description and Priority

The "register for competition" function should allow players to register for competitions through the competitions page in his/her personal accounts. Players should be able to view the competition schedule and, from there, pick their desired competition. Upon registration, players will be asked to fill out a form of the personal information required for registration and will be asked for any payments necessary to successfully complete registration. This function is of high priority, since competitions are a vital part of a sports club.

4.2.2 Stimulus/Response Sequences

In order to register for a competition, a player would have to click on the "view competition schedule" option, then he/she would proceed to select a competition and click on the "register for competition" option.

4.2.3 Functional Requirements

REQ-1: In order for this feature to function, the software must have access to a database with the competition schedule.

4.3 Checkout Equipment

4.3.1 Description and Priority

This function should allow players and coaches to checkout equipment. The users should be able to view a list of equipment along with the status: available or unavailable. After a checkout is complete, this function should also update the status of the piece of equipment that was just checked out to "unavailable". It should also register the checkout date and display the due date to the user. This function is of high priority as checking out equipment is essential for a sports club.

4.3.2 Stimulus/Response Sequences

To stimulate the behavior designed for this feature, the user would have to click on the "view equipment" option and then proceed to go through the list and select the desired piece of equipment.

4.3.3 Functional Requirements

REQ-1: In order for this feature to function, the software must have access to a database with all the equipment at the club along with their status.

REQ-2: Should a player/coach return any piece of equipment late, the function should display a late message and register a fine for the user accordingly.

4.4 Make Payment

4.4.1 Description and Priority

This function should allow players to make payments upon registering for classes, competitions, or memberships by providing their credit card and personal information. It should calculate the amount owed by the user and display it. It should also update the status of a payment to "complete" once it has successfully been completed. This function is of high priority, since making payments is essential for all types of registrations.

4.4.2 Stimulus/Response Sequences

This function should be activated whenever a player attempts to make any reservation that requires payment whether it is for classes, competitions, or new memberships.

4.4.3 Functional Requirements

REQ-1: In order for this feature to function efficiently and securely, security measures must be put in place, such as multi factor authentications and other methods that are TBD

REQ-2: In the case that a user enters an invalid credit card number, an error message should be displayed and the payment declined.

4.5 Login

4.5.1 Description and Priority

This function should allow players, coaches, and admin to login providing their username and password through the homepage. Upon logging in, they would be taken to their personal accounts. The user has 5 attempts to login, after which they will not be able to continue attempting to login. This function is of high priority because no personal account or information can be accessed without logging in.

4.5.2 Stimulus/Response Sequences

This function should be activated when a user attempts to login through the login page.

4.5.3 Functional Requirements

REQ-1: In order for this feature to function, the system must have a file with all usernames and corresponding passwords in order to compare the two during a login attempt.

REQ-2: In the case that a username and password do not match any of the existing login credentials, an "incorrect username/password" message should pop up.

REQ-3: After 5 login attempts, the system should block the user from attempting another login until a password reset is done.

4.6 Register as New Member

4.6.1 Description and Priority

This function should allow players to set up a new account by providing personal details such as name, last name, birthdate, username, password, email, and phone number. This function is of high priority because it is an essential step in increasing the number of new registrations for the club.

4.6.2 Stimulus/Response Sequences

This function should be activated whenever a new member selects the "sign up" option through the homepage.

4.6.3 Functional Requirements

REQ-1: In the case that a user enters a username that is already taken, the function should send an error message. The function should have access to a file with all existing usernames in order to do that.

REQ-1: In the case that a user leaves any item blank, the function should display an error message.

4.7 Display Team Stats

4.7.1 Description and Priority

This function should allow an administrator to access team statistics such as number of total wins, losses, and draws in the current season, and a graph of the number of wins per month. The administrator would access this information through his/her personal account through the team statistics page by providing the team name. This function is of low priority as it is

not a function that the club depends on to continue functioning smoothly or to ensure its financial success.

4.7.2 Stimulus/Response Sequences

This function should be activated whenever an administrator accesses the team statistics page on his/her personal account and attempts to search for a particular team.

4.7.3 Functional Requirements

REQ-1: In order for this feature to function, a file that stores all team games along with their dates and status (win, loss, draw) has to be maintained.

REQ-1: In the case that a user enters an invalid team name, an error message should be displayed.

4.8 Display Financial Report

4.8.1 Description and Priority

This function should allow an administrator to access a financial report through his/her personal account under the Finances page. The administrator would have to provide a time frame for which he/she wants the report. The function would display the number of new memberships made in that time frame, along with the revenue made from new memberships, class registrations, and competition registrations during that time frame. This function is of medium priority as it is not vital for the club to function, however, it is important to track the club's financial situation.

4.8.2 Stimulus/Response Sequences

This function should be activated whenever an administrator accesses the financial report option through the finances page.

4.8.3 Functional Requirements

REQ-1: In order for this feature to function, the system has to have access to files that store memberships (along with their dates), class registration, and competition registration.

REQ-1: In the case that a user enters an invalid time frame, an error message should be displayed.

4.9 Edit Schedule

4.9.1 Description and Priority

This function should allow coaches to edit their schedule through the My Schedule page on his/her personal account. The user would have to input the change he/she would like to

make and then click to confirm. The function should then update the schedule and display the new updated schedule to the user. A confirmation message should be displayed after a successful edit has been made. This function is of high priority because another function, the "register for class" function depends on it, since it depends on coaches' up-to-date schedules.

4.9.2 Stimulus/Response Sequences

This function should be activated whenever a coach clicks the "edit schedule" option through the My Schedule page on his/her personal account.

4.9.3 Functional Requirements

REQ-1: In order for this feature to function, the system has to have access to files that store all the coaches' schedules.

4.10 Cancel Class

4.10.1 Description and Priority

This function should allow players to cancel an existing booking of a class through their schedules on their personal accounts. The user would have to select the class he/she wants to cancel then click "cancel" to cancel class. The function should display a list of existing classes from which the user can pick from and it should also update that list after the cancellation. The function should also update the player's schedule accordingly. It should also display a confirmation message once the cancellation is completed successfully. This function is of high priority because canceling a certain class is one of the essential functions of registration.

4.10.2 Stimulus/Response Sequences

This function should be activated whenever a player accesses his/her class list through his/her personal account.

4.10.3 Functional Requirements

REQ-1: In order for this feature to function, the system has to have access to files that store a player's class list and schedule.

4.11 View Personal Schedule

4.11.1 Description and Priority

This function should allow players and coaches to view their personal schedules through the My Schedule page on their personal accounts. This function is of high priority, since it is important for the users to be able to access their up-to-date schedules whenever they may need.

4.11.2 Stimulus/Response Sequences

This function should be activated whenever a player or coach clicks on the My Schedule through his/her personal account.

4.11.3 Functional Requirements

REQ-1: In order for this feature to function, the system has to have access to files that store all players' and coaches' schedules.

4.12 Edit Competition Schedule

4.12.1 Description and Priority

This function should allow administrators to edit the competition schedule by adding or deleting competitions through the Competitions page on their personal accounts. To add a new competition the administrator would have to provide information such as the competition name, date, and description. To delete a competition, the administrator would have to simply select the competition and choose to delete it. This function should update the competition schedule. It is of high priority because another function "register for competition" depends on it.

4.12.2 Stimulus/Response Sequences

This function should be activated whenever an administrator clicks on the "edit competition schedule" option through the Competitions page.

4.12.3 Functional Requirements

REQ-1: In order for this feature to function, the system has to have access to files that store the competition schedule along with every competition's information.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system has to be efficient and capable of handling multiple requests in real-time.

Title: Hard Drive Space

Description: Storage space needed to store the database.

Required: At least 500 MB.

Ideally: No more than 1 GB.

Title: System's Memory Usage

Description: The amount of CPU memory the system needs to function properly.

Rationale: The amount of memory used should be observed during the system testing

Required: At least TBD MB

Ideally: No more than TBD MB.

5.2 Safety Requirements

The system should be designed to prevent data loss and system intrusions.

Title: Creating Account Safety

Description: Precautions when customers create an account.

Rationale: If a customer wants to create an account and the username the choose already exists then the customer will be asked to choose a different user name.

Requirement: The customer should not be allowed to register if the username already exists.

5.3 Security Requirements

Title: Communication Security
Description: The security of communication between the server and the system.
Rationale: The messages between the system and server should be encrypted to avoid leaking of credentials.
Requirement: All the messages sent from the log-in window should be encrypted.

Title: Account Security

Description: Security of any account in case of failure to logout.
Rationale: Any account needs to be automatically logged out after a certain period of inactivity.
Requirement: Every account should be logged out from the system after 12 minutes of inactivity.
Title: Failed Login Attempts
Description: In case there are a number of consecutive failed login attempts.
Rationale: The management should be notified of the failed login attempts in case there is an attempt to access the system from an unauthorized party.
Requirement: The IP address should not be able to attempt to log in for 1hour after 5 consecutive failed attempts.

5.4 Software Quality Attributes

Title: Search Accuracy

Description: How accurate is the search function in each data set.

Rationale: The system should show the right result on a search.

Requirement: The search result should be accurate at least 98% of the time.

Ideally: The search results will be accurate for every search.

Title: System Availability

Description: The reliability of the system when it is being used.

Rationale: The system should be available to be used when the user needs it.

Requirement: The system should run without a problem at least 98% of the time.

Ideally: The system should run every time with no issues

Title: Application flexibility

Description: The application should be flexible for editing. It should have simple coding and proper documentation in case developers need to edit the system or add new features

Rationale: In case there are changes that need to be implemented in the future.

Title: System compatibility

Description: The system should be compatible with different operating systems.

Rationale: The system should be able to run on Windows, Mac, IOS and android systems.

5.5 Business Rules

Club managers can keep track of equipment and their schedule.

Users can register and log into their accounts as well as make payments and register for classes.

Feasibility Study

- **Technical Feasibility**:

1) Users' and analysts' familiarity with the business area:

The business area is the registration and data management process in a sport club. The services that the system is going to offer are: Keeping track of inventory, producing financial reports, keeping track of coach's schedules and managing payments. The team's members have good information about this business area by being on the customer end of it which can help with finding and solving problems that they themselves and other customers experience. Also the team's members have worked with data managing projects before which adds to their experience and skill set with these types of business areas. However, this is the first time the team's members are working on a system for a sports club so they are not experts on it.

2) Familiarity with technology:

The technical tools we are going to use:

- The programming language is going to be C++.
- The Database Management System the team is going to use is SQL Server.
- Web language is going to be HTML.

Most of the team's members are familiar with C++ and SQL, while they are not experts HTML.

The team members can use personal computers to build this system therefore there is not going to be a need for any specific machine.

The system is going to need an online server to store all the data and a cloud storage needed to back up all the information.

Considering that the users in the sports club will be transitioning from a paper- based system, they may have to go through a short training in order to learn how to use all features of the system correctly.

3) Project Size:

The team is composed of four members and the duration of the project is approximately seven weeks.

4) Conclusion:

The risk in this stage is low. The team is familiar with both the technology and the business area of this project. Moreover, the cost of the project is moderately low which makes this project to be highly feasible.

System Size
Function Point Estimation

Functionality	Input	Output	Queries	File	Program Interface
Register for Class	2	1	2	3	0
Make Payment	1	2	1	1	1
Register for Competition	2	1	2	2	0
Edit Schedule	1	2	1	1	0
Login	1	2	1	2	0
Cancel Class	1	2	1	2	0
Checkout Equipment	1	1	1	2	0
Register as New User	1	1	1	2	0
Display Team Stats	1	4	1	1	0
Display Financial Report	1	4	0	3	0
View Personal Schedule	0	0	1	1	0
Edit Competition Schedule	2	2	2	2	0

	Complexity				
Description	**Total #**	**Low**	**Medium**	**High**	**Total**
Inputs	14	11 * 3	2 * 4	1 * 6	47
Outputs	22	16 * 4	6 * 5	0 * 7	94
Queries	14	14 * 7	0 * 10	0 * 15	98
Files	22	19 * 7	3 * 10	0 * 15	163
Program Interface	1	0 * 5	1 * 7	0 * 10	7
Total Unadjusted Function Point (TUFP) =					**409**

- **Total Processing Complexity (PC):-**

Complexity is rated from 0 to 3: (0 = no effect on project complexity; 3 = great effect on project complexity).

Tasks	Complexity (0-3)
Data Communication	3
Team Cohesion	1
Familiarity with Technology	3
On-line Data Entry	2
Total Processing Complexity (TPC) =	9

- **Adjusted Processing Complexity (APC):-**

APC = 0.65 + (0.01*TPC)
APC = 0.65 + (0.01 * 9) = 0.74

- **Total Adjusted Function Points (TAFP):-**

TAFP = TUFP *APC
TAFP = 409 * 0.74 = 302.66

- **Converting Function Points to Lines of Code (LOC):-**

Language/ Tool	Number of LOC / FP
Python	21
HTML	15
JavaScript	15
SQL	13

Percentages:-
- 25% will be done in Python
- 25% will be done in HTML
- 25% will be done in JavaScript
- 25% will be done in SQL

- **Number of Lines of Code (LOC)**

LOC = TAFP * #(LOC/FP) * %
For Python = 302.66 * 21 * 0.25 = 1588.97
For HTML = 302.66 * 15 * 0.25 = 1134.98
For JavaScript = 302.66 * 15 * 0.25 = 1134.98

For SQL = 302.66 * 13 * 0.25 = 983.65

Total LOC = 4842.58

- **Estimated Effort:-**

Effort = 2.4 * LOC/1000
Effort = (2.4 * 4842.58)/1000
= 11.62 person months

- **Estimated Schedule Time:-**

Time = 2.5 * (effort)^0.38
Time = 2.5 * (11.62)^0.38
= 6.35 months

- **Estimated Number of Persons:-**

Average # of persons = effort/time
= 11.62/6.35
= 1.83 persons

- **Estimated Time with 4 Persons (Group Size):-**

Time = effort/persons
= 11.62/4
= 2.91 months

Economic Feasibility

Costs	**Period 1**	**Period 2**	**Period 3**	**Period 4**	**Period 5**
Salaries	100	100	100	100	0
H/W & S/W	50	0	0	0	0
Training	20	10	0	0	0
Support & Maintenance	0	0	0	0	5
Total Costs (DHS)	170	110	100	100	5

Benefits	**Period 1**	**Period 2**	**Period 3**	**Period 4**	**Period 5**
Increase in # of Customers	0	0	0	0	1000
Decrease in Costs	0	0	0	0	10
Total Benefits (DHS)	0	0	0	0	1010
NCF	-170	-110	-100	-100	1005
CNCF	-170	-280	-380	-480	525

ROI (%)	5.16
BEP (%)	0.48
Project Time (Days)	135

Period 6	Total
0	400
0	50
0	30
5	10
5	490

Period 6	Total
2000	3000
10	20
2010	3020
2005	2530
2530	5060

Software Process Model

I think the spiral model is ideal for this project because of its iterative nature and because the requirements of the project are well understood. Since we came up with the system and we know what we want, we can go through the process of communication, planning, modeling, construction, and deployment. However, we also want to get feedback after every iteration, so we can determine if we want to make any new changes or if something didn't work out the way we intended. This will give the system more flexibility while still able to complete all of the requirements that were initially agreed upon.

Gantt Chart