

SPORTS CLUB MANAGEMENT SYSTEM

Submitted for the partial fulfilment of the requirements

For the award of the degree in

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

BY

EDWIN AROKIA RAJ E

(DEPT NO: 16-UCS-136)

Under the Guidance of

Prof. I. JUSTIN SOPHIA M.Sc., M.Phil.,



DEPARTMENT OF COMPUTER SCIENCE

LOYOLA COLLEGE (AUTONOMOUS)

CHENNAI – 600034

APRIL-2019

BONAFIDE CERTIFICATE

This is to certify that the project work entitled “**SPORTS CLUB MANAGEMENT**” is being submitted to Loyola College (Autonomous), Chennai-600034 by **EDWIN AROKIARAJ E(16-UCS-136)** for the partial fulfilment for the award of degree of Bachelor of Computer Applications is a Bonafide record of work carried out by him , under my guidance and supervision.

HEAD OF THE DEPARTMENT
Dr. C.MUTHU M.sc.,M.Tech.,Ph.D

PROJECT GUIDE
Prof.I.JUSTIN SOPHIA M.Sc..M.Phil.,

The Viva-Voce Examination held on _____ at Loyola College (Autonomous), Chennai – 600034.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

Sports event management is to provide which manages the activity of many sports at a time. It also manages the selection activity of students to college/school and to state level. This application will provide the information regarding a sports tournament and will provide all necessary details about the tournament. The user will consume less amount of time when compared to manual paper work through the automated system. The system will provide the serving activity in quick and easy manner. To maintain information about the teams participating in the tournament and related to particular sports. The main idea of the project is to organize the sport events under state level and district level. The student's information will be saved in the database depending on the particular sports. Tournament venue and date will be maintained in the database. Sports teacher and also sports students can use this system. The winner in the particular event will announced through this system. The selected students for the next level will also announced using this system.

There are two roles ADMIN and USER. Admin has all the access and he can create match with other team, View match details and update the match results.

User can register, Login pay the fees, apply for a match which was created by admin and can see the match results.

ACKNOWLEDGEMENT

First of all, I would like to thank the Almighty God without whom this task would have not been achieved. I express my sincere and profound gratitude to our college principal of Loyola College **Rev. Dr F. Andrew, S.J.** and Deputy Principal **Dr. Fatima Vasanth** for their encouragement and support to do my project in our computer Lab.

I would also like to thank our Head of the Department **DR.C.MuthuM.SC.,M.Tech.,PH.D.,**and I would also like to thank our co-ordinator **Prof.Ms.M.Regina M.Sc.,M.C.A.,M.Phil.,** for their encouragement to finish this project.

My special thanks to my guide **Prof. I. Justin Sophia M.Sc., M.Phil.,** who gave me the needed information and support for the successful completion of this project.

I take this opportunity to thank all the staff members and the computer science department who rendered their help directly to finish my project in time.

Last but not the least, I would like to express my hearty thanks to my parents, without whom I would not have come to this level in my life. My hearty thanks to my friends and well-wishers who supported and encouraged me to complete this project successfully.

TABLE OF CONTENTS

S.NO	CONTENTS	PAGE NO
1.	INTRODUCTION	1
2.	SYSTEM ANALYSIS	2
	2.1 Problem Definition	2
	2.2 Existing System	2
	2.3 Proposed System	3
	2.4 System Requirements	4
	2.4.1 Software Requirements	4
	2.4.2 Hardware Requirements	4
3.	SYSTEM DESIGN	5
	3.1 Architectural Design	5
	3.2 Database Design	9
	3.3 GUI Design	12
4.	PROJECT DESCRIPTION	16
5.	SYSTEM DEVELOPMENT	21
	5.1 Language/Tool	21
	5.2 Pseudo Code	22
6.	SYSTEM TESTING AND VALIDATIONS	25
	6.1 Test Cases and Testing	26
	6.2 Validations	30
7.	USER MANUAL	36
8.	SYSTEM IMPLEMENTATION	43
9.	CONCLUSION	49
10.	FUTURE ENHANCEMENT	50

11.	BIBLIOGRAPHY	51
	APPENDIX B– SAMPLE CODING	53
	APPENDIX C– SAMPLE OUTPUT	58

Chapter 1

Introduction

1.General

A club management system project that provides and manages various club activities such as member registration, registration for various regular and vacation batches and more. The sports club management system software is a .net built system that manages the entire club activities and provides respective functionality for various types of visitors. This system is built with respect to managing a cricket club. It allows normal users to avail for club membership, book the ground at for desired days and even enroll for various activities in the club. The sports club management system is built keeping in mind various various daily activities of a cricket club and the software automates all these club functionality for easy operation of the club.

2.Organization Profile (optional) :

Chapter 2

System Analysis

1.Problem Definition :

In all the current sports clubs are have their students records in file or paper works and also the fees payment will be done with paper work, will be leads more difficulty to maintain users/students records and it will need lots of human efforts. Possible of human error also there in current sports club management systems. Also all the match records will be stored in database for future reference.

Our system will overcome all these flows so that each and every student record can be maintain in database and also there not be any human error and mismatch of student or match details.

Performance Requirements

The following performance characteristics should be taken care of while developing the system:

- **User friendliness**

The system should be easy to learn and understand so that new user can also use the system effectively, without any difficulty.

- **User satisfaction**

The system should meet user expectations.

- **Error handling**

Response to user errors and the undesired situations should be taken care of to ensure that the system operates without halting.

- **Response time**

The response time of all the operations should be low. This can be made possible by careful programming.

- **Safety**

The system should be able to avoid or tackle catastrophic behavior.

- **Robustness**

The system should recover from undesired events without human intervention.

2.Existing System :

In the existing Sports Event Management system, students are not able to get proper information about the games conducted in various colleges. The student needs to spend the time to get the information about the game. The student should attend the venue to get registered for the game which takes a lot of time.

Disadvantages

- Student cannot view upcoming match details created by admin.
- Students wont have any dashboard to view their progress in that sports club.
- Students cannot pay through online.
- Admin cannot update the completed match details.

3.PROPOSED SYSTEM:

The software to be produced is on “Sports Management System (SMS)”. There are 2 users’ i.e., the admin and the student. Admin provides username and password to the students. He also has the right to add or modify the given username and password of the student. Using this username and password, an students can login to the system. The first procedure is the student registration. Here the students enters all the details of a student including the sport they like to participate. All these information will be stored in the database. Next is the entry of the tournaments. Here depending on the sport, the tournament date and the venue is saved in the database for further confirmation. There are 5 different sports. A list of players will be displayed and also the upcoming tournaments will be shown according to the particular sport. The system also provides a special authority of adding photos to the system for a particular sport for sweet memories and for proof.

The system also does the selection procedure to the college level and the state level competition. To the college level all the players are allowed to play. Whereas to the state level, the best players will be selected from the college level list which will be also saved in the database.

Advantages

- It becomes convenient for members to view completed and upcoming match details
- Admin can create a new match against any team.
- Admin can update the completed match details and score card.
- The system calculates proper cost calculations for expected period hence there does not arise any need of negotiating for cost.
- The system stores and sends all the members details to the admin for requests.
- The system is easy to use and saves human efforts and cost.

4. System Requirement

HARDWARE REQUIREMENTS

- Processor : I3 or above
- RAM : 4 GB
- Hard Disk : 250 GB

SOFTWARE REQUIREMENTS

- Operating System : Windows 7 or Above

TECHNOLOGIES

- Xampp
- PHP
- Mysql 5.0
- Java 1.8
- JavaScript

Chapter - 3

System Design

1. Architectural Design

Architectural Designing is the most important phase of software development. It requires a careful planning and thinking on the part of the system designer. Designing software means to plan how the various parts of the software are going to achieve the desired goal. It should be done with utmost care because if the phase contains any error then that will effect the performance of the system, as a result it may take more processing time, more response time, extra coding workload etc.

Software design sits at the technical kernel of the software engineering process and is applied regardless of the software process model that is used. After the software requirements have been analyzed and specified, software design is the first of the three technical activities Designing, Coding and Testing that are required to build and verify the software. Each activity transforms information in such a manner that ultimately results in validated computer software.

DESIGN GOALS

The following goals were kept in mind while designing the system:

- ***Make system user-friendly.*** This was necessary so that system could be used efficiently and system could act as catalyst in achieving objectives.
- ***Make system compatible i.e.*** It should fit in the total integrated system. Future maintenance and enhancement must be less.
- ***Make the system compatible*** so that it could integrate other modules of system into itself.
- ***Make the system reliable, understandable and cost-effective.***

Data Flow Diagram Of The System

The Data Flow Diagram shows the flow of data or information. It can be partitioned into single processes or functions. Data Flow Diagrams can be grouped together or decomposed into multiple processes.

The DFD is an excellent communication tool for analysts to model processes and functional requirements. Used effectively, it is a useful and easy to understand modeling tool. It has broad application and usability across most software development projects. It is easily integrated with data modeling, workflow modeling tools, and textual specs. Together with these, it provides analysts and developers with solid models and specs. Alone, however, it has limited usability. It is simple and easy to understand by users and can be easily extended and refined with further specification into a physical version for the design and development teams.

The different versions are Context Diagrams (Level 0, Level 1...), Partitioned Diagrams (single process only -- one level), functionally decomposed, and leveled sets of Data Flow Diagrams.

Data Store

A repository of information. In the physical model, this represents a file, table, etc. In the logical model, a data store is an object or entity.

Data Flows

DFDs show the flow of data from external entities into the system, showed how the data moved from one process to another, as well as its logical storage. There are only four symbols:

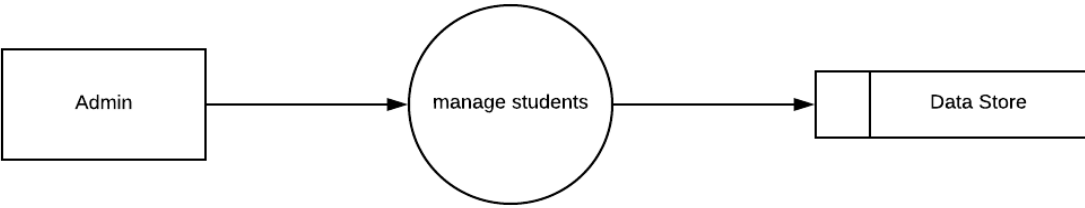
1. Squares representing external entities, which are sources or destinations of data.
2. Rounded rectangles representing processes, which take data as input, do something to it, and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items.
4. Open-ended rectangles representing data stores

There are several common modeling rules for creating DFDs:

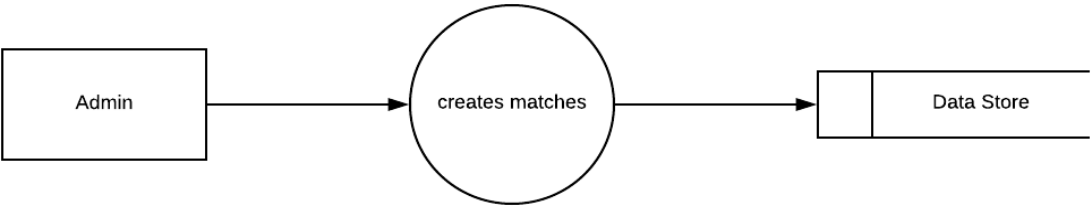
1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.
5. A data flow must be attached to at least one process.

Data Flow Diagram

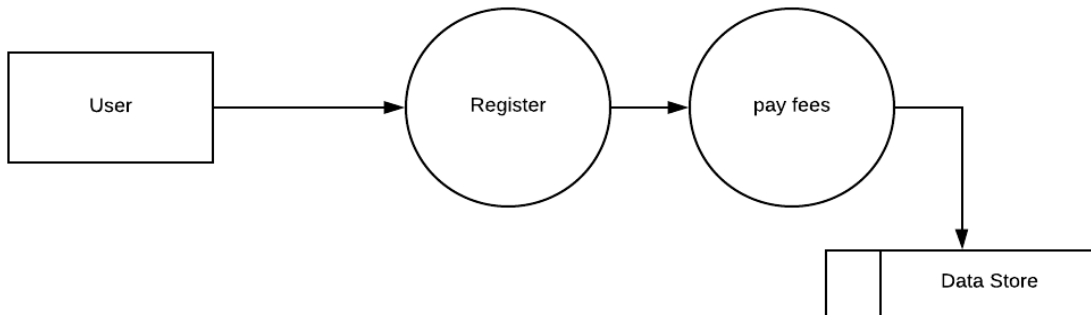
LEVEL 0



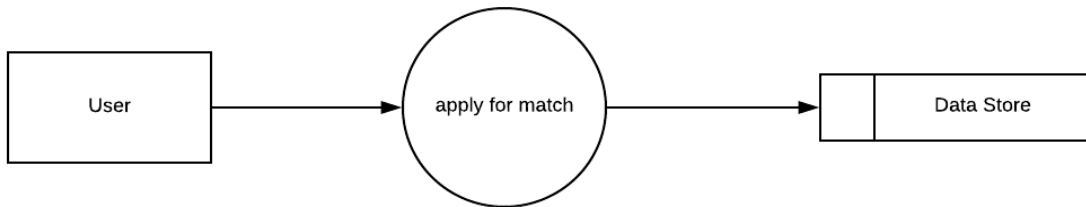
LEVEL 1



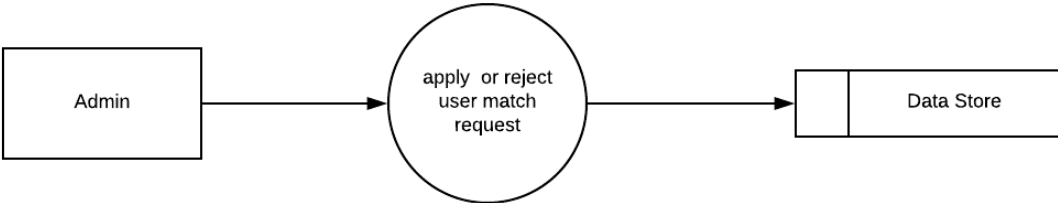
LEVEL 2.1



LEVEL 2.2



LEVEL 2.3



2.Database Design

We used MySQL database for database connection and used 4 tables to store user , match and match result details . Each table have its index (Primary Key) so that there is no possible for the data reputation and also each table have relationship between other table through foreign key. Following table structures shows each table description used for our Sports Club Management System.

Match table

This table contains the information about the various details and rate for each match created by the admin. In this table match id is primary key with non-null index so that each and every records should be unique to maintain non-reputation match records.

Name	type	length	null	constraints
match_id	int	20	Not null	Primary key
Match name	varchar	200	Not null	
ground	varchar	200	Not null	
date	date		Not null	
Opp_team	varchar	200	Not null	

User table

This table contains the information about the various user details of students name username password etc. . In this table user_id name is primary key with non-null index so that each and every records should be unique to maintain non-reputation students records.

Name	type	length	null	constraints
User_id	Int	20	Not null	Primary key
Username	varchar	20	Not null	
pass	varchar	10	Not null	
email	varchar	20	Not null	

match table

This table contains the information about the various match details of match name, opponent name, ground name etc. . In this table match_id name is primary key with non-null index so that each and every records should be unique to maintain non-reputation match records.

Name	type	length	null	constraints
match_id	Int	20	Not null	Primary key
Match name	varchar	200	Not null	
date			Not null	
ground	varchar	200	Not null	
opponent	varchar	200	Not null	

User match table

This table contains the information about the various match attend by a student such as match name, status, scores . In this table match_id name is primary key with non-null index so that each and every records should be unique to maintain non-reputation match records.

Name	type	length	null	constraints
match_id	Int	20	Not null	Primary key
Match name	varchar	200	Not null	
Status	varchar	200	Not null	
Our score	varchar	200	Not null	
Opponent score	varchar	200	Not null	

Entity Relationship (ER) Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

ER diagram has three main components:

- 1.Entity
- 2.Attribute
3. Relationship

1. Entity

An entity is an object or component of data. An entity is represented as rectangle in an



ER diagram.

2. Attribute

An attribute describes the property of an entity. An attribute is represented as Oval in an ER



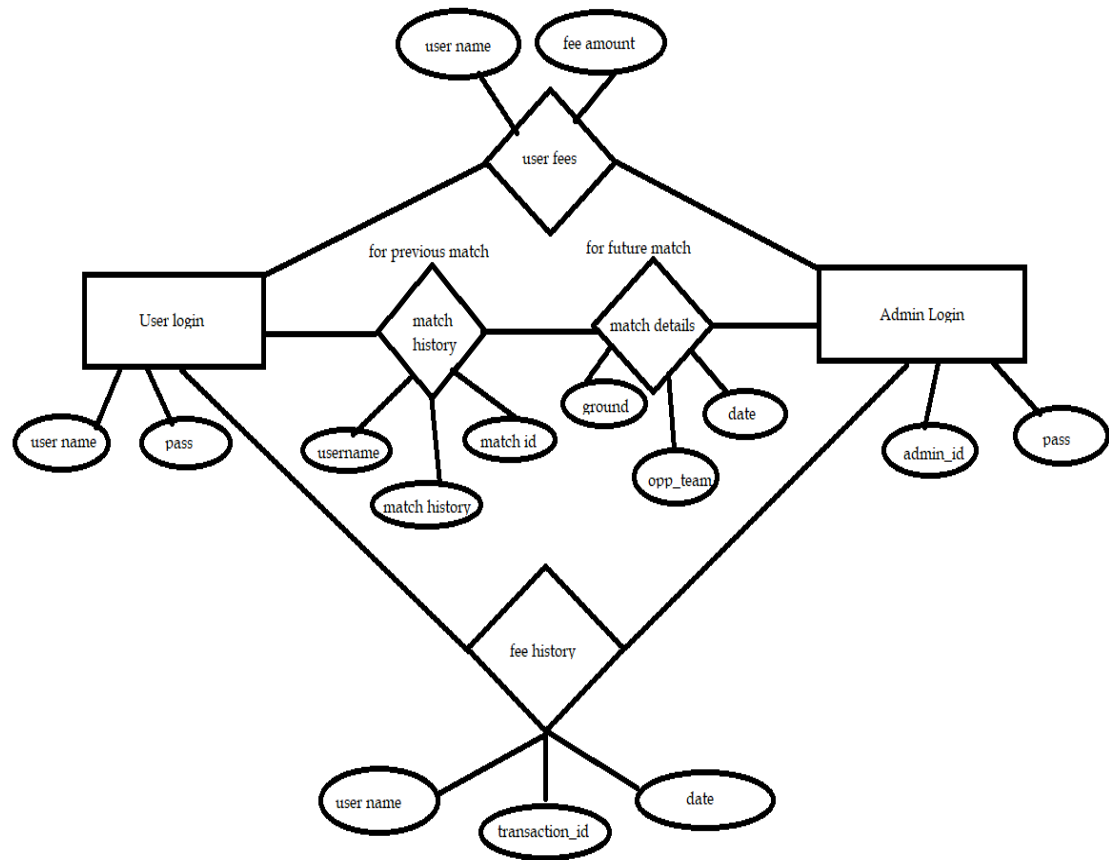
diagram

3. Relationship



A relationship is represented by diamond shape in ER diagram

The Following ER diagram shows the entity relationship between each table and its attributes. Through this diagram we can clearly see the primary key and foreign key constraints for each tables.



Chapter – 4

Project Description

The project brings the entire manual process of sports event management online which is built using PHP as a front end and SQL Server as a back end. The main purpose of this project is to simplify the process of handling each sports event by providing a web interface for admin and teacher. The admin part consists of multiple modules to initiate with the sports event by adding the type of sport (indoor or outdoor), adding student who are interested in a particular sports activity, adding teachers who will conduct the particular sports activity which is allotted by the admin itself and lastly, viewing the results of sports event held in college. The teacher part has come up with handling all the sports related activity assigned by the admin. Teacher performs various task such as taking the attendance of the students who are registered for a particular sport event, viewing the list of students to mark the winner of each round, generating the results based on multiple rounds won by the student and also can view the 1st, 2nd and 3rd standings of student's name for the particular sport event.

ADVANTAGES

- Overcomes the dependency of a single person handling all the sport activity.
- Multiple teacher can use the web interface to login and perform the desired task.
- Easy results generation and view rankings.
- Easy to access the system anywhere and anytime.

Chapter – 5

System Development

1.Language / Tool

XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. The Apache HTTP Server, colloquially called Apache, is the world's most used web server software. MariaDB database - MariaDB is a community-developed fork of the MySQL relational database management system intended to remain free under the GNU GPL. It's made by the original developers of MySQL and guaranteed to stay open source. Notable users include Wikipedia, WordPress.com: Create a website or blog and Google. There are interpreters for scripts written in the PHP and Perl programming languages. It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

PHP

PHP stands for Hypertext Preprocessor (no, the acronym doesn't follow the name). It's an open source, server-side, scripting language used for the development of web applications. By scripting language, we mean a program that is script-based (lines of code) written for the automation of tasks.

What does open source mean? Think of a car manufacturer making the secret to its design models and technology innovations available to anyone interested. These design and technology details can be redistributed, modified, and adopted without the fear of any legal repercussions. The world today might have developed an amazing supercar!

Web pages can be designed using HTML. With HTML, code execution is done on the user's browser (client-side). On the other hand, with PHP server-side scripting language, it's executed on the server before it gets to the web browser of the user.

PHP can be embedded in HTML, and it's well suited for web development and the creation of dynamic web pages for web applications, e-commerce applications, and database applications. It's considered a friendly language with abilities to easily connect with MySQL, Oracle, and other databases.

JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities

- Immediate feedback to the visitors – They doesn't have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.
- Less server interaction – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

SSL

SSL: Is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral.

To be able to create an SSL connection a web server requires an SSL Certificate. When you choose to activate SSL on your web server you will be prompted to complete a number of questions about the identity of your website and your company. Your web server then creates two cryptographic keys - a Private Key and a Public Key.

The Public Key does not need to be secret and is placed into a Certificate Signing Request (CSR) - a data file also containing your details. You should then submit the CSR. During the SSL Certificate application process, the Certification Authority will validate your details and issue an SSL Certificate containing your details and allowing you to use SSL. Your web server will match your issued SSL Certificate to your Private Key. Your web server will then be able to establish an encrypted link between the website and your customer's web browser.

MySQL

MySQL is an Oracle-backed open source relational database management system based on Structured Query Language. MySQL runs on virtually all platforms, including Linux, Windows .

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

2.Pseudo code

Pseudo code is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

Pseudo code is an artificial and informal language that helps programmers develop algorithms. Pseudo code is a "text-based" detail (algorithmic) design tool. The rules of Pseudo code are reasonably straightforward. All statements showing "dependency" are to be indented

Chapter – 6

System Testing and Validations

Software testing is a critical element of the ultimate review of specification design and coding. Testing of software leads to the uncovering of errors in the software functional and performance requirements are met. Testing also provides a good indication of software reliability and software quality as a whole. The result of different phases of testing are evaluated and then compared with the expected results. If the errors are uncovered they are debugged and corrected. A strategy approach to software testing has the generic characteristics:

- ❖ Testing begins at the module level and works “outwards” towards the integration of the entire computer based system.
- ❖ Different testing techniques are appropriate at different points of time.
- ❖ Testing and debugging are different activities, but debugging must be accommodated in the testing strategy

Goals and Objectives

“Testing is a process of executing a program with the intent of finding an error”. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Our Objective is to design test processes that systematically uncover different classes of errors and do so with minimum amount of time and effort.

Statement of scope

A description of the scope of the software testing is developed. All the features to be tested are noted as follows. The basic principles that guides software testing are,

- ❖ All test cases should be traceable top customer requirements. The most severe defects from the customer's point of view are those that cause the program to fail to meet its requirements.
- ❖ Test case should be planned long before testing begins. Testing plan can begin as soon as the requirement model is complete. Detailed definition of the test cases can begin as soon as the design is solidified. Therefore, the entire test can be planned before any code has been generated.
- ❖ Testing should begin “in the small” and progress towards “in the large”. The first test planned and executed generally focus on the individual modules. As testing progresses testing shifts focus in an attempt to find errors in integrating clusters of modules and ultimately in the entire system

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. Testing is a set of activities that can be planned in advance and conducted systematically. For this reason a template for software testing, a set of steps into which we can place specific test case design techniques and testing methods should be defines for software process.

Testing often accounts for more effort than any other software engineering activity. If it is conducted haphazardly, time is wasted, unnecessary effort is expanded, and even worse, errors sneak through undetected. It would therefore seem reasonable to establish a systematic strategy for testing software.

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The unit testing is normally considered as an adjunct step to coding step. Because modules are not a standalone program, drivers and/or stubs software must be developed for each unit. A driver is nothing more than a “main program” that accepts test cases data and passes it to the module. A stub serves to replace the modules that are subordinate to the modules to be tested. A stub may do minimal data manipulation, prints verification of entry and returns.

Approaches used for Unit Testing were:

1. **Functional Test:** Each part of the code was tested individually and the panels were tested individually on all platforms to see if they are working properly.

Performance Test: These determined the amount of execution time spent on various parts of units and the resulting throughput, response time given by the module.

2. **Stress Test:** A lot of test files were made to work at the same time in order to check how much workloads can the unit bear.
3. **Structure Test:** These tests were made to check the internal logic of the program and traversing particular execution paths.

2. INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunication. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

INTEGRATION TESTING TECHNIQUES

1. WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basic path testing:

Flow graph notation

Cyclometric complexity

Deriving test case

Graph matrices Control

2. **BLACK BOX TESTING**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

1. • Graph based testing methods
2. • Equivalence partitioning
3. • Boundary value analysis
4. • Comparison testing

3.ACCEPTANCE TESTING

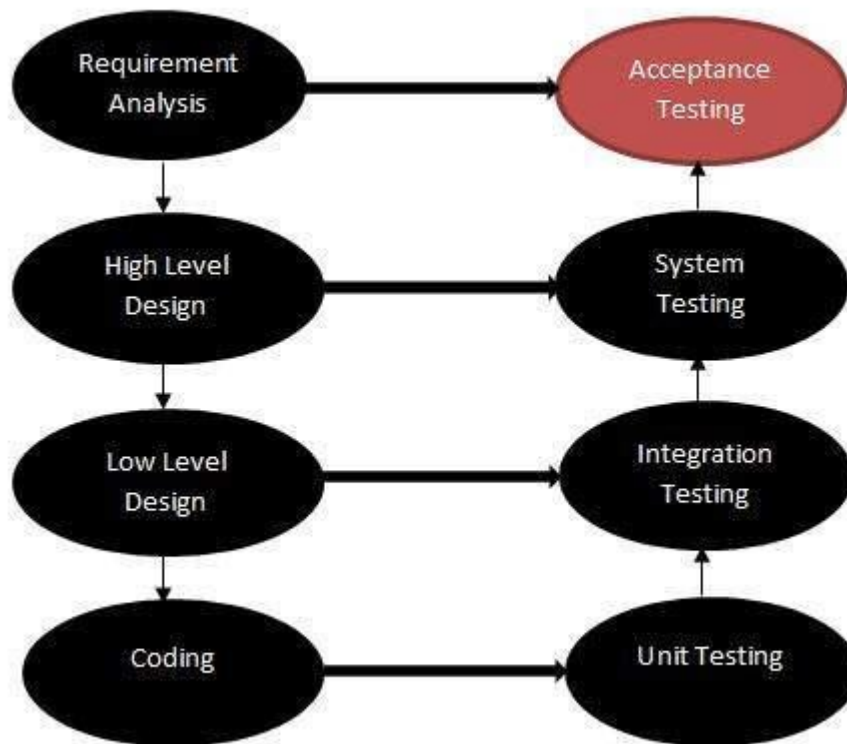
Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing
- Alpha Testing
- Beta Testing

Acceptance Testing - In Software Development Life Cycle

The following diagram explains the fitment of acceptance testing in the software development life cycle.



4. VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

Chapter – 7

User Manual

User Manual

In the system there are mainly one user interface using which a user is able to

- Login
- Register
- Admin Page
- Add new match
- Update Match
- View User/Match records
- User Page
- Apply For a match
- View completed match details

Following are the features available in the project:

Admin:

- Manage Admin User.
- Approve Register Customer.
- Mange Games.
- Manage Team.
- Manage Player.
- Manage Schedule.
- Manage Tournament.
- View Booking and confirm it.

User

- Request for Registration.
- Login to Portal.
- Manage Profile.
- View Tournament.
- View Schedule.
- View Team Details.
- Apply for a upcoming match

Modules of the software:

- Login Registration: Admin provides username and password to the employee. He also has the right to add or modify the given username and password of the employee. Using this username and password, an employee can login to the system.
- Student Registration: The first procedure is the student registration. Here the employee enters all the details of a student including the sport they like to participate. All these information will be stored in the database.
- Tournament: Here depending on the sport, the tournament date and the venue is saved in the database for further confirmation.
- Games: . A list of players will be displayed and also the upcoming tournaments will be shown according to the particular sport. The system also provides a special authority of adding photos to the system for a particular sport for sweet memories and for proof.
- Selection procedure: The system also does the selection procedure to the college level and the state level competition. To the college level all the players are allowed to play. Whereas to the state level, the best players will be selected from the college level list which will be also saved in the database.
- Logout: This module allows the user to Logout the application. Further operations cannot be performed after user exits

Chapter – 8

System Implementation

Once the system was tested, the implementation phase started. A crucial phase in the system development life cycle is successful implementation of new system design. Implementations simply mean converting new system design into operation. This is the moment of truth the first question that strikes in every one's mind that whether the system will be able to give all the desired results as expected from system. The implementation phase is concerned with user training and file conversion.

The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of computer system. Implementation is used here to mean the process of converting a new or revised system design into an operational one. Conversion is one aspect of implementation. The other aspects are the post implementation review and software maintenance. There are three types of implementation:

- ❖ Implementation of a computer system to replace a manual system
- ❖ Implementation of a new computer system to replace an existing one.
- ❖ Implementation of a modified application to replace an existing one.

Conversion

Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks and personnel irritation to a minimum. It involves creating computer compatible files; training the operational staff; installing terminals and hardware. A critical aspect of conversion is not disrupting the functioning of organization.

Direct Implementation

In direct implementation; the previous system is stopped and new system is started up coincidentally. Here there is a direct change over from manual system to computer-based system. In direct change over implementation; employess can face the problems. Suppose our software is not working much efficiently as manual one then we can't find the defects in our software. It will not be beneficial in finding errors.

Chapter – 9

Conclusion

The objective of this project was to build a program for maintaining the details of all students and match details for a sports club. “Sports Management System” project which helps students to save a lot of time in searching for games being conducted in various colleges. Our project provides students to get register from anywhere and anytime. The system developed is able to meet all the basic requirements. It will provide the facility to the user so that they can keep tracks of all the students performance. The management of the whole club and the student will be also benefited by the proposed system, as it will automate the whole supply procedure, which will reduce the workload. The security of the system is also one of the prime concerns.

There is always a room for improvement in any software, however efficient the system may be. The important thing is that the system should be flexible enough for future modifications. The system has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly.

1. **Goal achieved:** The System is able provide the interface to the user so that he can replicate his desired data. .
2. **User friendliness:** Though the most part of the system is supposed to act in the background, efforts have been made to make the foreground interaction with user as smooth as possible. Also the integration of the system with Inventory Management project has been kept in mind throughout the development phase.

Chapter – 10

Future Enhancement

We have successfully finished our sports club management System. But we would like to improve our system in future. Our current system is website based. We want to turn this system into android application so that it will be more user friendly. Also nowadays security is of major importance. So we would like to tighten up the security of our system. Finally we would like to add few options for editing the information that are already saved into the our system. We sincerely hope that our System will help all to work effectively with their students and matches for their easy.

So, Finally we would like to do the following future enhancement in our system so that it will work more effectively

- **Improvement in Security**
- **Convert Same as android mobile Application**
- **Edit Option for available for stored match and student details**

Chapter - 11

Bibliography

1. Appendix A - Data Dictionary

Data Dictionary is a simply repositories to store information about all data items. A data dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management systems keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.

User: Administrator / student

Rite Of Security : Administrator

Data Bases: football

Tables:

- user table
- match table
- user match table

Files:

- login.php – for login page
- match_new.php – for create new match
- admin.php – admin console
- pay_fees.php – fees payment page
- register_match.php – registering for new match
- user.php – user console

Data Items:

User id
name
email
password
Match id
Match name
Date
Ground name
Opponent name
Match status
Our score
Opponent score

2. Appendix B – Code Sample

index.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Footbal</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <div class="header_div">
      <marquee>
        <h2>Football club management system</h2>
      </marquee>
    </div>
    <div class="navbar">
      <ul>
        <li><a class="active" href="index.php">Home</a></li>
        <li><a href="about.php">About</a></li>
        <li><a href="login.php">Login</a></li>
        <li><a href="register.php">Register</a></li>
        <li><a href="logout.php">Logout</a></li>
      </ul>
    </div>
    
  </body>
</html>
```

register.php

<?php

```
include("db_connection.php");
```

```
if(isset($_POST['fname'])){
```

```
    $f_name = $_POST['fname'];
```

```
    $email = $_POST['emai'];
```

```
    $pass1 = $_POST['psw1'];
```

```
    $pass2 = $_POST['psw2'];
```

```
    $fee_amount = $_POST['fee'];
```

```
    if($fee_amount != '500'){
```

```
        $alert_amount_not_correct = "<script>alert('Initial fees should be 500 Rs')</script>";
```

```
        echo $alert_amount_not_correct;
```

```
    }else if($pass1 != $pass2){
```

```
        $alert_password_mismatch = "<script>alert('Given Passwords are mismatch')</script>";
```

```
        echo $alert_password_mismatch;
```

```
    }else{
```

```
        $ins = "INSERT INTO user_table(fullname, email, password, admin) VALUES  
('f_name','$email','$pass1','0')";
```

```
        $re = mysqli_query($db,$ins);
```

```
        if($re){
```

```
            $last_id = mysqli_insert_id($db);
```

```
            $digits = 10;
```

```
            $tran_id = rand((int)pow(10, $digits-1), (int)pow(10, $digits)-1);
```

```
            $insert_fees_details = "INSERT INTO fees_table(transaction_id, user_id, user_name,  
amount) VALUES ('$tran_id','$last_id','$f_name','$fee_amount')";
```

```
            $re2 = mysqli_query($db,$insert_fees_details);
```

```
if($re2){
    header("Location: user.php");
}
}else{
    $alert_fail = "<script>alert('Registration Fail')</script>";
    echo $alert_fail;
}
}
}
```

?>

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Footbal</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="header_div">
        <marquee>
            <h2>Football club management system</h2>
        </marquee>
    </div>
    <div class="navbar">
        <ul>
            <li><a class="active" href="index.php">Home</a></li>
            <li><a href="about.php">About</a></li>
            <li><a href="login.php">Login</a></li>
            <li><a href="register.php">Register</a></li>
```

```
<li><a href="logout.php">Logout</a></li>
</ul>
</div>

<h2>Register Form</h2>
<form autocomplete="off" action="#" method="POST">
  <div class="form_container">
    <label for="fname"><b>Full Name</b></label>
    <input class="input_fields" type="text" name="fname" required>
    <label for="emai"><b>Email</b></label>
    <input class="input_fields" type="email" name="emai" required >
    <label for="psw1"><b>Password</b></label>
    <input class="input_fields" type="password" name="psw1" required>
    <label for="psw2"><b>Re-enter Password</b></label>
    <input class="input_fields" type="password" name="psw2" required>
    <label for="fee"><b>Initial Fess : <span style="color:red">Rs.500/-</span></b></label>
    <input class="input_fields" name="fee" type="number" required>
    <button type="submit">Register</button>
  </div>
</form>
<br>
<br>
<a href="login.php">Already register?</a>
</body>
</html>
```

register_match.php

```
<?php
include("db_connection.php");

if(isset($_POST['match_id'])){
    $user_id = $_SESSION['user_id'];
    $match_id = $_POST['match_id'];
    $f_name = $_SESSION['fullname'];

    $in = "INSERT INTO user_match(match_id, user_id, full_name) VALUES
('$match_id','$user_id','$f_name')";
    $re = mysqli_query($db,$in);
    if($re){
        $alert_sess = "<script>alert('Successfully Registered for this match')</script>";
        echo $alert_sess;
    }else{
        $alert_fail = "<script>alert('Register Fail')</script>";
        echo $alert_fail;
    }
}

?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Footbal</title>
    <link rel="stylesheet" href="css/style.css">
</head>
```

```

<body>
<div class="header_div">
    <marquee>
        <h2>Football club management system</h2>
    </marquee>
</div>
<div class="navbar">
    <ul>
        <li><a class="active" href="index.php">Home</a></li>
        <li><a href="about.php">About</a></li>
        <li><a href="login.php">Login</a></li>
        <li><a href="register.php">Register</a></li>
        <li><a href="logout.php">Logout</a></li>
    </ul>
</div>

```

```

<h2>Register For A match</h2>

```

```

<?php
$sel = "SELECT * FROM `match_table` WHERE status='New'";
$re = mysqli_query($db,$sel);
if(mysqli_num_rows($re) > 0){
    while($ro = mysqli_fetch_assoc($re)){
        echo "<h3><span style='color:red'>Match Name : </span>".$ro['name'].</h3>";
        echo "<h3><span style='color:red'>Date : </span>".$ro['date'].</h3>";
        echo "<h3><span style='color:red'>Time : </span>".$ro['time'].</h3>";
        echo "<h3><span style='color:red'>Ground : </span>".$ro['ground'].</h3>";
        echo "<h3><span style='color:red'>Opponent : </span>".$ro['opponent'].</h3>";
        echo '<form action="#" method="POST">
            <input type="hidden" name="match_id" value="'.$ro['match_id'].'">

```

```

        <input type="submit" value="Apply For This Match">
    </form>';
    echo "<hr>";
}
}else{
    echo '<h1 style="color:red">No Result Found</h1>';
}
?>
<br>
<br>
<a href="user.php">user main page</a>
</body>
</html>

```

update_match.php

```

<?php
include("db_connection.php");

if(isset($_POST['mat_res'])){
    $m_id = $_GET['m_id'];
    $mat_res = $_POST['mat_res'];
    $our_score = $_POST['our_score'];
    $opp_score = $_POST['opp_score'];

    $up    =    "UPDATE    match_table    SET    status    =    '$mat_res',our_score    =
'$our_score',opponent_score = '$opp_score' WHERE match_id = '$m_id'";

```



```
$re = mysqli_query($db,$up);
if($re){
    $alert_update_success = "<script>alert('Update Success')</script>";
    echo $alert_update_success;
}else{
    $alert_update_fail = "<script>alert('Update Fail')</script>";
    echo $alert_update_fail;
}
}
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Footbal</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="header_div">
        <marquee>
            <h2>Football club management system</h2>
        </marquee>
    </div>
    <div class="navbar">
        <ul>
            <li><a class="active" href="index.php">Home</a></li>
            <li><a href="about.php">About</a></li>
            <li><a href="login.php">Login</a></li>
            <li><a href="register.php">Register</a></li>
            <li><a href="logout.php">Logout</a></li>
        </ul>
```

```

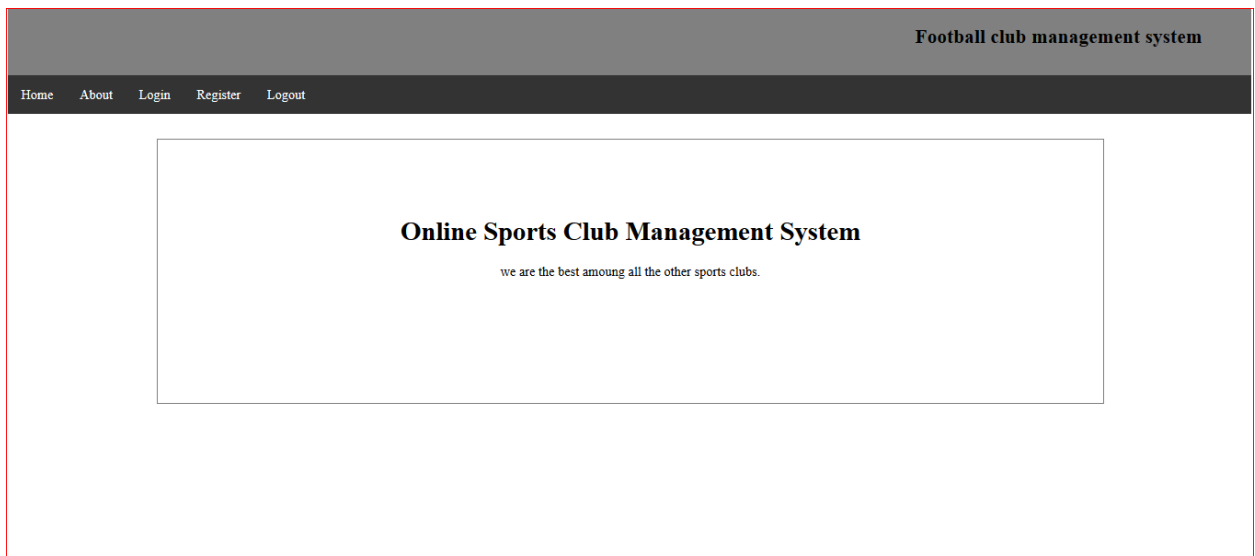
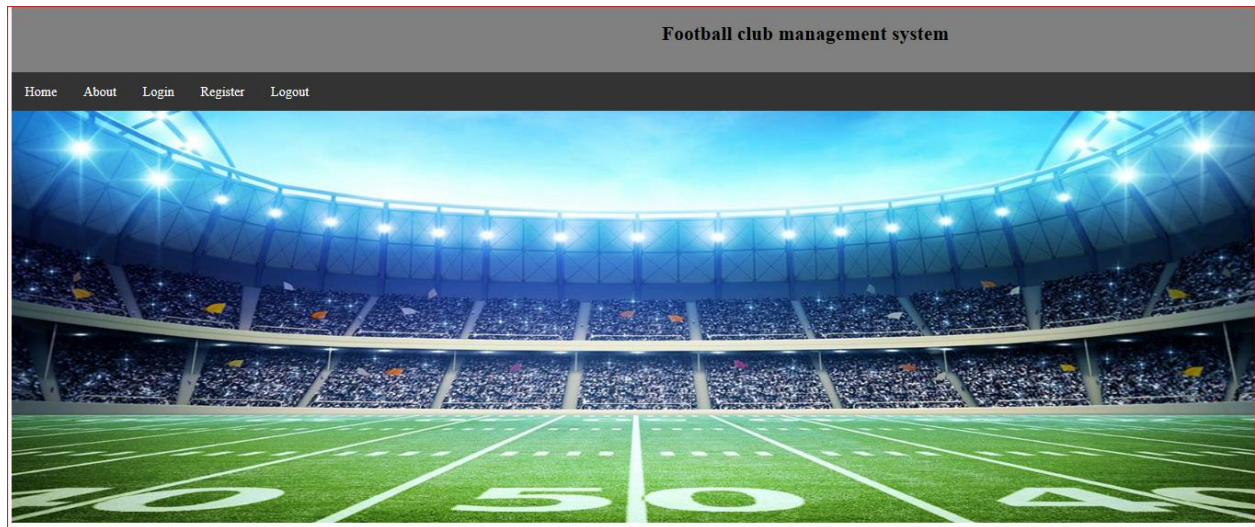
</div>
<h2>Update match details</h2>
<br>
<?php
    if(isset($_GET['m_id'])){
        $match_id = $_GET['m_id'];
        $se = "SELECT * FROM `match_table` WHERE match_id = '$match_id'";
        $re = mysqli_query($db,$se);
        if(mysqli_num_rows($re) > 0 ){
            $ro = mysqli_fetch_assoc($re);
            $match_id = $ro['match_id'];
            echo "<h3><span style='color:red'>Match Id : </span>".$ro['match_id'].</h3>";
            echo "<h3><span style='color:red'>Match Name : </span>".$ro['name'].</h3>";
            echo "<h3><span style='color:red'>Date : </span>".$ro['date'].</h3>";
            echo "<h3><span style='color:red'>Time : </span>".$ro['time'].</h3>";
            echo "<h3><span style='color:red'>Ground : </span>".$ro['ground'].</h3>";
            echo "<h3><span style='color:red'>Opponent : </span>".$ro['opponent'].</h3>";
        }else{
            $alert_fail = "<h1 style='color:red'>No result found</h1>";
            echo $alert_fail;
        }
    }
?>
<form action="#" method="POST">
    <div class="form_container">
        <label for="mat_res"><b>Match Result</b></label>
        <select class="input_fields" name="mat_res" required>
            <option value="">Select Result</option>
            <option value="Won">Won</option>
            <option value="Loss">Loss</option>
        </select>
    </div>
</form>

```

```
<label for="our_score"><b>Our Team Score</b></label>
<input class="input_fields" type="number" name="our_score" required>
<label for="opp_score"><b>Opponent Team Score</b></label>
<input class="input_fields" type="number" name="opp_score" required>
<button type="submit">Update</button>
</div>
</form>
<br>
<a href="admin.php">admin home</a>

</body>
</html>
```

3. Appendix C - Sample Output screenshots



Football club management system

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Login Form

Email

Password

Login

[New User Register Here](#)

Football club management system

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Register Form

Full Name

Email

Password

Re-enter Password

Initial Fess : **Rs.500/-**

Register

[Already register?](#)

Football club management system

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Hi Charles Francis

Loss match count = 1

Won match count = 1

[Register for a new match](#)

[View my match results](#)

[View all match results](#)

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Hi Charles Francis

Your Match results

Match Name : CPL

Date : 2019-03-15

Time : 10:10

Ground : Sims Football Ground

Opponent : DFG Football Team

Result : Loss

Score Card : Our Team - 10 | 15 - DFG Football Team

Match Name : samson

Date : 2020-04-08

Time : 07:59

Ground : Vels Football Ground

Opponent : V5 Football Team

Result : Won

Hi Charles Francis

Your Match results

Match Name : CPL

Date : 2019-03-15

Time : 10:10

Ground : Sims Football Ground

Opponent : DFG Football Team

Result : Loss

Score Card : Our Team - 10 | 15 - DFG Football Team

Match Name : samson

Date : 2020-04-08

Time : 07:59

Ground : Vels Football Ground

Opponent : V5 Football Team

Result : Won

Hi Charles Francis

View Match Form

Enter match name

Match Id : 4

Match Name : test

Date : 2019-03-08

Time : 10:27

Ground : Gopalapuram Corporation play ground

Opponent : DFG Football Team

Result : Won

Score Card : Our Team - 5 | 3 - DFG Football Team

Football club management system

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Admin Console

[Search for a user](#)

[Create New Match](#)

[View All Match Details](#)

Football club management system

[Home](#)[About](#)[Login](#)[Register](#)[Logout](#)

Search user Form

Enter user name

Search

User Id : 2

Fullname : samson

Email : sam@gmail.com

Fees : paid

[admin home](#)

Football club management system

[Home](#) [About](#) [Login](#) [Register](#) [Logout](#)

New Match Form

Set a name for match

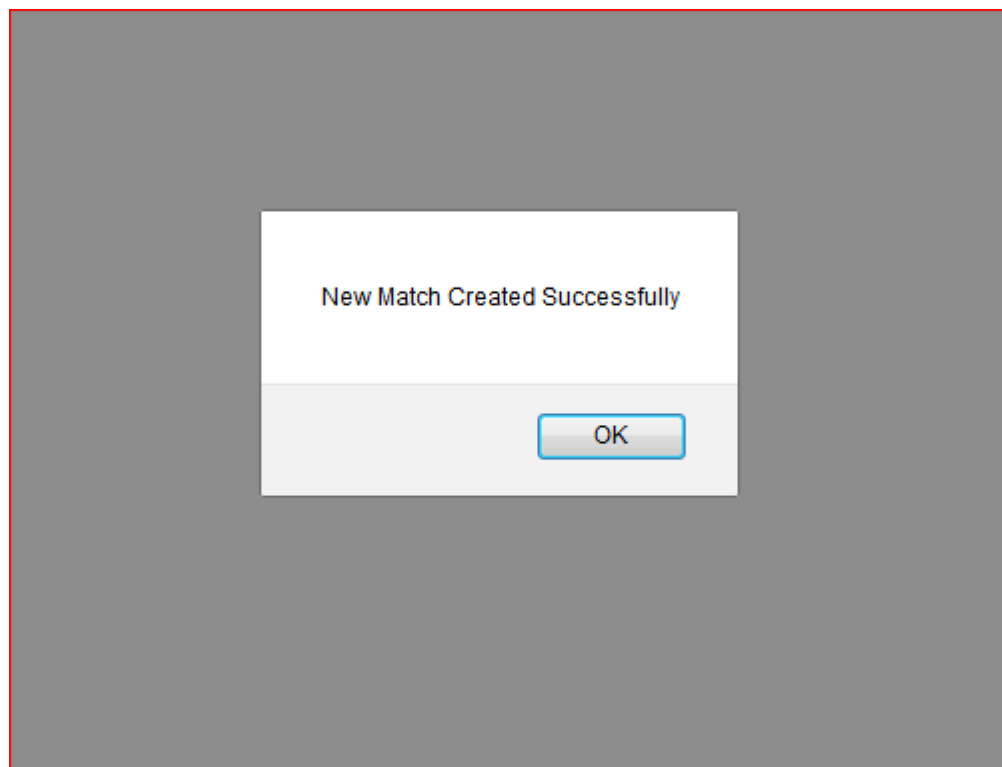
Date

Time

Ground

Opponent Team

[admin home](#)



View Match Form

Enter match name

Match Id : 1

Match Name : footbal match

Date : 2019-03-08

Time : 11:00

Ground : CMS Football Ground

Opponent : DFG Football Team

Result : Won

Score Card : Our Team - 10 | 4 - DFG Football Team

4.Appendix D - Reference

Books Referred

- “Mlearning PHP”
- “RATE ME IN PHP”- Sams Pearson Education [Lowell Mauer]

Sites Referred

- www.w3schools.com
- www.clubmanage.com
- www.vbforums.com/sports club management.php?p=2686697
- www.final-yearprojects.co.cc