MINI PROJECT

FOOTBALL LEAGUE MANAGEMENT SYSTEM

PROJECT REPORT

IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION

SUBMITTED BY:

KISHAN LAL P T

REG NO: -140021042789

UNDER THE GUIDANCE OF

Mrs. CHIPPY BABY

2014-2017

DEPARTMENT OF COMPUTER SCIENCE

C.E.T. COLLEGE OF MANAGEMENT, SCIENCE & TECHNOLOGY

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

N.P THOMAS VALLEY, AIRAPURAM

PIN: 683541

C.E.T. COLLEGE OF MANAGEMENT SCIENCE & TECHNOLOGY, AIRAPURAM

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

CERTIFICATE

This is to certify that this project work entitled "**FOOTBALL LEAGUE MANAGEMENT SYSTEM**" is a bonafide work done by "**KISHAN LAL P T**", Reg No: 140021042789 submitted in partial fulfilment for the degree of **Bachelor of Computer Application** of **MAHATMA GANDHI UNIVERSITY, KOTTAYAM** during the academic year 2014-2017.

INTERNAL GUIDE	**H.O.D**	**PRINCIPAL**

SUBMITTED FOR THE UNIVERSITY VIVA-VOICE EXAMINATION HELD ON …….....

Internal Examiner External Examiner

C.E.T. COLLEGE OF MANAGEMENT SCIENCE & TECHNOLOGY, AIRAPURAM

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**FOOTBALL LEAGUE MANAGEMENT SYSTEM**" submitted to **MAHATMA GANDHI UNIVERSITY** in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Computer Application** is a record of the original work done by "**KISHAN LAL P T**" under my supervision and guidance and that this project work has not formed the basis for the award of any Degree/Diploma/Fellowship or similar title to any candidates of any University.

PRINCIPAL **INTERNAL GUIDE**

COLLEGE SEAL **H.O.D**

SUBMITTED FOR THE UNIVERSITY VIVA-VOICE EXAMINATION HELD ON …….....

INTERNAL EXAMINER

DECLARATION

I KISHAN LAL P T hereby declare that the mini project work entitled "**FOOTBALL LEAGUE MANAGEMENT SYSTEM**" submitted in the partial fulfilment of the requirement for the award of **Bachelor Of Computer Application** of **M.G University**, Kottayam is a report of original work done by me during the period of study at C.E.T. College Of Management, Science And Technology, Airapuram under the supervision and guidance of **Mrs.Chippy Baby** Department Of Computer Science.

Place: Airapuram **KISHAN LAL P T**

Date:

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgements, then words play the heralding role of expressing my gratitude.

I thank the almighty for giving me the courage and enthusiasm to complete this project. I wish to convey my sense of gratitude to, **Prof. Paul Thomas,** M.Tech, M.B.A, M.Com the Principal, C.E.T. College of Management, Science & Technology.

I would like to extend my sincere thanks to **Mrs. Poornima N,** H.O.D of Computer Science Department, C.E.T. College of Management, Science & Technology for the support provided during the period of completion of this project.

I would like to show my eternal gratitude by thanking my class tutor **Mrs. Kamila Nizam,** Assistant Professor, Department Of Computer Science, for her motivation and guidance throughout our college life and especially during this project duration.

I would like to express my sincere thanks and appreciations to my internal guide **Mrs. Chippy Baby**, Assistant Professor, Department Of Computer Science, for the timely and intelligent guidance throughout the project period.

Words cannot express my deep sense of love, respect and gratitude to my dear and near ones for their wisdom, encouragement, support and care throughout this time.

KISHAN LAL P T

ABSTRACT

Football League Management System is a system that will be used to manage football competition by using league system in the tournament. Basically, the system will be designed to suit with the current football tournament. The system will be developed using VB and SQL as the database. It will be focused on managing all the information used during the tournament started from registration process, during the tournament and until the tournament is over. Currently most of football organization use filing method to store and manage all the information during the tournament period. The filing method gives few complications for staff to handle all those data. The system that is going to be developed will provide a complete systematic system which will maintain all the tournament data and come out any reports. Therefore, all the jobs during the tournament took part will become easier and manageable.

CONTENTS

LIST OF TABLES

1. INTRODUCTION

1.1 INTRODUCTION

The technological revolution has caused a boom in the field of computers. With the development of computer software's much tedious work for workers has become a simple job. An overall change took place in the functions of organization by the launching of computers. They started a growing awareness for a need of computerization.

Taking into consideration the present scenario, the project "FOOTBALL LEAGUE MANAGEMENT SYSTEM" is designed to handle the material details efficiently. This system is windows based and user friendly. The software can sustain in future development. This system is fully automated with four modules as,

1. Player Management
2. Team Management
3. Match
4. Stadium

1.2 OBJECTIVES OF THE PROJECT

Our objective is to make software with a high performance, the software which is easy to work with. The software is also able to extract details from database and make report easily. Unauthorized access to this software is blocked. Data redundancy problem is erased. To verify the validity of data check constraints are provided. Last but not the least, the most important of all objectives is to consider user satisfaction. The main advantages of the new systems are:

- Easy handling of certain Team and players.
- Helps in maintain the scheduling details.
- Helps in keeping the details of team, players, sponsors etc.

2. SYSTEM ANALYSIS

2.1 IDENTIFICATION OF NEEDS

System analysis refers to an orderly structured process for identifying and solving problem using computer. It is the most essential part of the project development. It is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvement to the system.

Training, experience and common sense are required for the collection of information needed to do the analysis. To analysis a system one has to study the system work in detail, before designing the appropriate computer based system that will meet the requirements of the system. In other words, system analysis specifies what the system analyst specifies what the system should do.

2.2 EXISTING SYSTEM

In the existing system, FOOTBALL LEAGUE MANAGEMENT SYSTEM is done as live. In this system, player management is a tedious task. Keeping all this in mind we have a developed a new system which addresses all the issues prevailing in the existing system and eliminated each and every one of them. By computerizing the existing system, all the hurdles faced by the collection and tedious task of searching a player is eliminated.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

The existing system has various disadvantages compared to the proposed system. Some disadvantages are listed below:

- Slow process of Registration a player to a Team.
- Less data integrity.
- Low in security.
- The processing of data was slow.

2.3 PROPOSED SYSTEM

Proposed system is fully computerized one. It does absolutely nothing on paper. As all the entire working is done with the aid of computers, this will result in faster processing and improved performance. This information can be quickly by the click of a button when compared to earlier retrieval from files and registers. As all the details are stored in the computer's hard drive the physical storage space requirement can be reduced to a great extent.

Processing of information manipulation was difficult with manual system, but as we are doing all the processing with the aid of computers it can be done very easily and accurately. Creating reports timely needs a quick search in the huge file cabinets, but with the proposed system reports can be generated in the winks of an eye lid.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Easy to record.
- Easy to access.
- Reduced man power.
- User interface.
- Faster access information.
- Efficient trace ability.
- Access large data in low time.
- Duplication of data will be avoided.
- Menu driven interface provides ease of use.
- Availability of previous data for future reference.

2.4 FEASIBILITY STUDY

All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. A feasibility study is not warranted for systems in which economic justification is obvious, technical risk is low, few legal problems are expected and no reasonable alternative exists. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be

cost effective from the business point of view and if it can be developed in the given existing budgetary constraints. The feasibility study should be relatively cheap and quick. The result should inform the decision of whether to go ahead with a more detailed analysis.

Feasibility study may be documented as a separated report to higher officials of the top-level management and can be included as an appendix to the system specification. Feasibility and risk analysis is related in many ways. If there is more project risk then the feasibility of producing the quality software is reduced. The study is done in these phases

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Behavioural feasibility

Technical feasibility

This is related to the technicality of the project feasibility if check the cost to conduct a full system investigation, cost of hardware and software. A system that can be developed technically and that will be used, if installed, must be still good.

Economic feasibility

The FOOTBALL LEAGUE MANGEMENT SYSTEM software supports the economic feasibility to a great extends. The cost of posting players, applying for a sponsorship, development of the system and the cost of hardware and software are not high. This reduces effort and time of us. This makes software economically feasible.

Always the financial benefits must be equal or exceed the cost. Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system or more commonly known as cost or benefits analysis.

Operational feasibility

Proposed systems are beneficial only if they can be turned into information systems. That is it will meet the organizations operating requirements and also checks that whether the system will work when it is developed and installed. Therefore, it is understandable that the introduction of a candidate system requires special efforts to educate, sell and train others.

The FOOTBALL LEAGUE MANGEMENT SYSTEM software supports the operational feasibility to a great extends. The performance of this software is more accurate, more user friendly, effective, error free.

Behavioural Feasibility

Computers have been known to facilitate changes where as people are intently resistant to change therefore it is necessary that an evaluation should be made about the user's attitude towards the new system. This is called behavioural feasibility.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 FRONT END

VISUAL BASIC 6 is a high-level programming language that evolved from the earlier DOS version called BASIC. BASIC means Beginners All-purpose Symbolic Instruction Code. Now, there are many versions of Visual Basic are available in the market, the latest being Visual Basic 2015.

Visual basic is an ideal programming language for developing sophisticated professional applications for Microsoft windows. The code looks a lot like English Language. It makes use of graphical user interface for creating robust and powerful applications. Coding in GUI environment is easy and quicker as compare to traditional, linear programming languages.

Features of visual basic

- GUI Interface
- Modularization
- Object Oriented
- Debugging
- Data Access Feature
- Macros IDE
- Studio IDE

GUI Interface:

VB is a Graphical User Interface language. This means that a VB program will always show something on the screen that the user can interact with to get a job done.

Modularization:

It is considered good programming practice to modularize your programs. Small modules where it is clearly indicated what comes into the module and what goes out makes a program easy to understand.

Object Oriented:

Object Oriented Programming is a concept where the programmer thinks of the program in "objects" that interact with each other. Visual Basic forces this good programming practice.

Debugging:

Visual Basic offers two different options for code debugging Debugging Managed Code Runtime Debugger. The Debugging Managed Code individually debugs C and C++ applications and Visual Basic Windows applications. The Runtime Debugger helps to find and fix bugs in programs at runtime.

Data Access Feature:

By using data access features, we can create databases, scalable server-side components for most databases, including Microsoft SQL Server and other enterprise-level database.

3.2 BACK END

SQL server 2005 is the native data store of c#.NET every business enterprise maintains large volumes of data for its operations. With more and more people accessing data for their work, the need to maintain its integrity and relevance increases. Normally with the traditional method of storing data and information in the files, the chances that the data losses, its integrity and validity are very high. There are however limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using mem-cached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters

3.3 HARDWARE REQUIREMENTS

The selection of hardware configuration is a very important task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient capacity to solve the database and the application. The network should be efficient to handle the communication fast.

Processor :- Intel Pentium IV / AMD, Running at 1GHz

RAM :- 2GB

Hard Disk :- 160 GB

Keyboard :- Standard keyboard

Monitor :- 15 Inch.

Mouse :- Two or Three Button Mouse

3.4 SOFTWARE REQUIREMENTS

Front-End Development Tool:- vb6

Database Server :- SQL server 2008

Operating system :- windows 7 and above

4. DATA FLOW DIAGRAM

4.1 INTRODUCTION TO DATA FLOW DIAGRAM

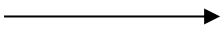
A Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by using a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles and are outside the system such as vendors or customers with whom the system interacts. They either supply or consume data. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number.

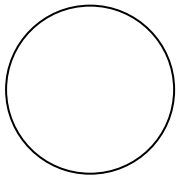
DFDs can be hierarchically organized, which help in partitioning and analysing large systems. As a first step, one Data Flow Diagram can depict an entire system, which gives the system overview. It is called Context Diagram of level 0 DFD. The Context Diagram can be further expanded. Thus, successive expansion of a DFD from the Context Diagram to those giving more details is known as levelling of DFD. Thus, a top down approach is used, starting with an overview and then working out the details.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the inputs (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user's requirements. The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

Basic Data Flow Diagram Symbols

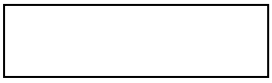


A **data flow** is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrowhead pointing in the direction of the flow.



Circles stand for process that converts data into information. A process represents transformation where incoming data flows are changed into outgoing data flows.

A **data store** is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.

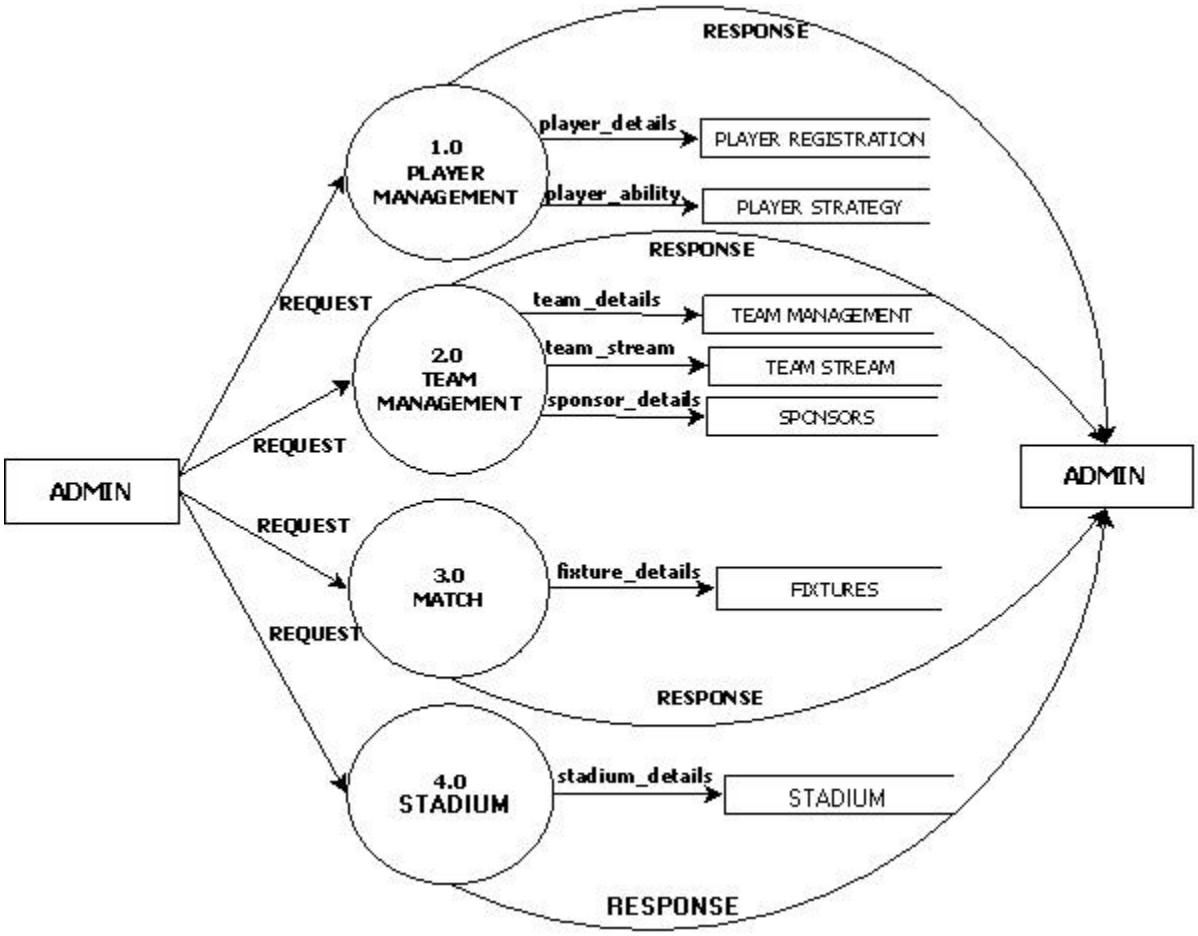


A **source or sink** is a person or part of organizations, which enter or receives information from the system, but is considered to be outside the contest of data flow model.

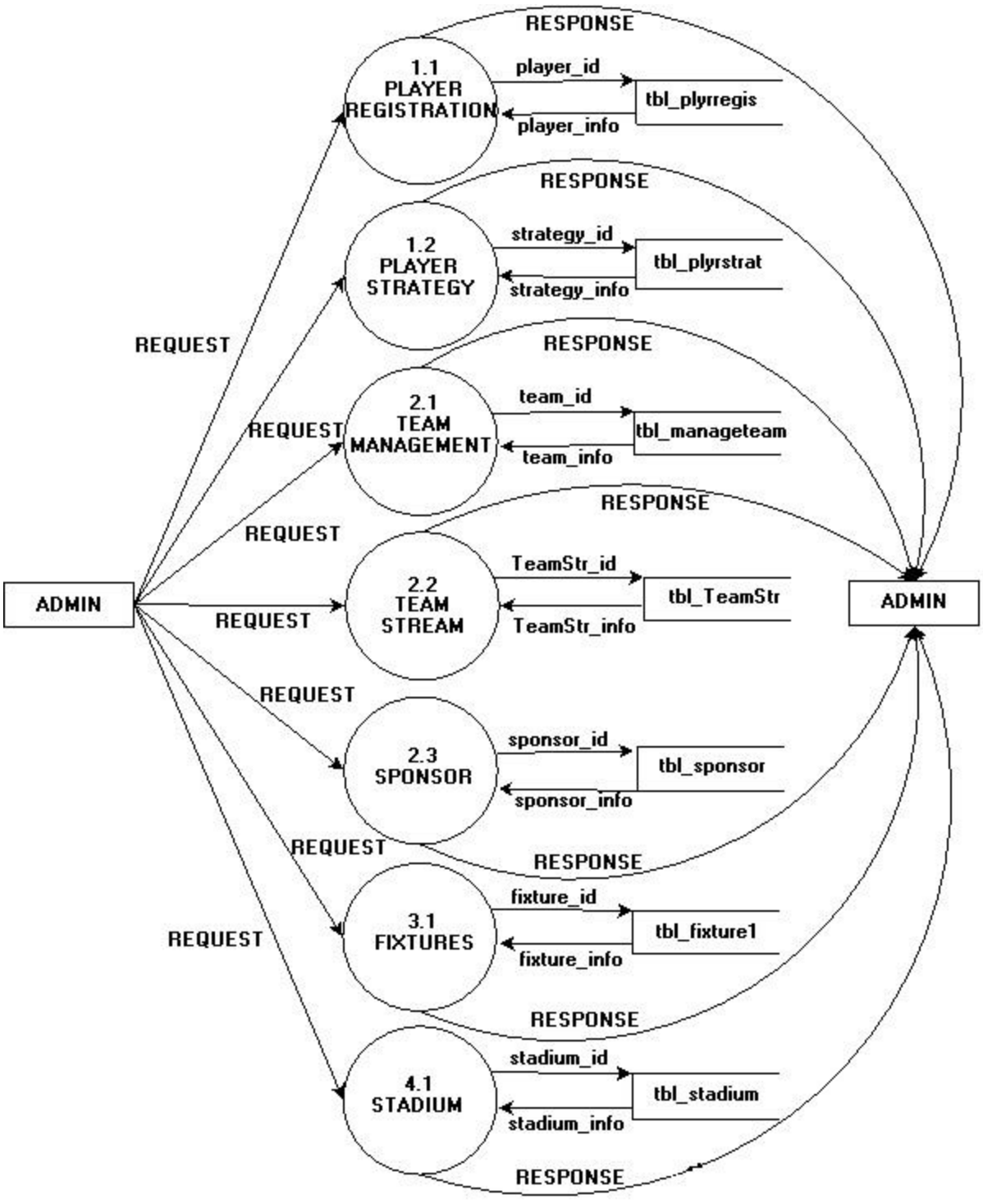
4.2 CONTEXT DIAGRAM

Level 0

Level 1


RESPONSE
1.0
PLAYER
MANAGEMENT
player_details
PLAYER REGISTRATION
player_ability
PLAYER STRATEGY
RESPONSE
REQUEST
2.0
TEAM
MANAGEMENT
team_details
TEAM MANAGEMENT
team_stream
TEAM STREAM
sponsor_details
SPONSORS
REQUEST
ADMIN
ADMIN
REQUEST
3.0
MATCH
fixture_details
FIXTURES
REQUEST
RESPONSE
4.0
STADIUM
stadium_details
STADIUM
RESPONSE

Level 2


ADMIN
REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
1.1
PLAYER
REGISTRATION
player_id
player_info
tbl_plyrregis
RESPONSE
1.2
PLAYER
STRATEGY
strategy_id
strategy_info
tbl_plyrstrat
RESPONSE
2.1
TEAM
MANAGEMENT
team_id
team_info
tbl_manageteam
RESPONSE
2.2
TEAM
STREAM
TeamStr_id
TeamStr_info
tbl_TeamStr
RESPONSE
2.3
SPONSOR
sponsor_id
sponsor_info
tbl_sponsor
RESPONSE
3.1
FIXTURES
fixture_id
fixture_info
tbl_fixture1
RESPONSE
4.1
STADIUM
stadium_id
stadium_info
tbl_stadium
RESPONSE
ADMIN

5. SYSTEM DESIGN

5.1 INTRODUCTION TO SOFTWARE DESIGN

System design is the creative act of invention developing new inputs, a database offline files, method, procedures and output for processing business data to meet organization objectives. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from user-oriented document to a document oriented to the programmers or database personnel. Characteristics of a well-designed system are:

- Accessibility
- Decision making ability
- Economy
- Flexibility
- Reliability
- Simplicity

The design will determine the success of the system. System design is based on the information gathered during system analysis. System design goes through two phases of development.

- Logical Design – DFD shows the logical flow of a system and defines the boundaries of the system. For the candidate system, it describes the inputs, outputs, databases and procedures-all in a format that meets the user's requirements.
- Physical Design - This produces the working system by defining the design specification that tells programmers exactly what the candidate system must do.

5.2 INPUT DESIGN

Input design is the link that ties the information system into the world of its users. The input design involves determining what the inputs are, how the data should be performed, how to validate data, how minimize data entry and how to provide a multi-user facility. Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. Input design is the process of converting user-originated inputs to a computer-based format. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing.

All the input data validated in the order and if any data violates any condition, the user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate tables in the database.

Log In Form

Players Registration

Players Strategy

Sponsors Registration

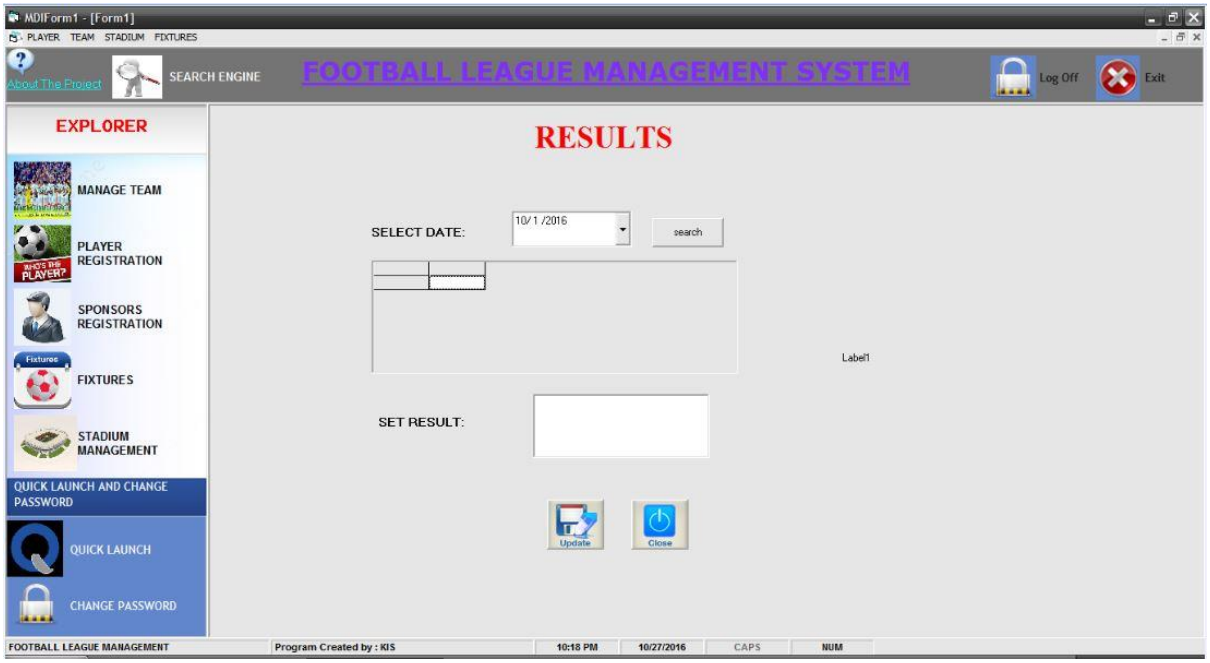
5.3 OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Output design is a very important phase because the output needs to be in an attractive manner. Efficient, intelligible output design improves the system relationship with the user and help in decision making. A major form of the output is the hard copy from the printer and screen reports. Printouts are designed around the output requirements of the user. Allowing the user to view to the sample screen is important because the user is ultimate judge of the quality of output. The output model of this system is the user-friendly window. These user-friendly windows are meant for the purpose of easy view of the stored information.

View Sponsors

Fixtures

Results



5.4 DATABASE DESIGN

A database is a collection of data. Database design refers to the design of the tables used to store data. The database involves name of records, data item with its name, type and size.

In the design of the database program first we have to thoroughly look into the requirements of the program for the design of database. Then we have to design how much tables are required in the database. Thereafter as per requirement of the end users we can decide which fields that must be in this table. As per a general rule a provision must be taken in the design for the future enhancement of the program. Some of important tables are:

Login:

Table: tbl_login

FIELDS	DATATYPE/SIZE	DESCRIPTION
user_name	varchar(50)	User name
pass_word	varchar(50)	Password

Manage Team:

Table: tbl_manageteam

Primary key : team_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
team_id	bigint	Team ID
team_name	varchar(50)	Team name
Place	varchar(50)	Team Place
Status	varchar(50)	Team status
Value	varchar(50)	Team value
stadium_name	varchar(50)	Stadium name
year_found	varchar(50)	Year found

Team Stream:

Table: tbl_TeamStr

Primary key:- TeamStrID

Foreign key:- team_ID

FIELDS	DATATYPE/SIZE	DESCRIPTION
TeamStrID	bigint	Team stream ID
Team_ID	varchar(50)	Team ID
management	varchar(50)	Management score
stability	varchar(50)	Stability score
support	varchar(50)	Support score
stadium	varchar(50)	Stadium score

Player Registration:

Table: tbl_plyrregis

Primary key:- player_id

Foreign key:- team_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
player_id	bigint	Player ID
first_name	varchar(50)	First name
last_name	varchar(50)	Last name
dob	datetime	Date of birth
team_id	varchar(50)	Team ID
previous_team	varchar(50)	Previous team
country	varchar(50)	Country
state	varchar(50)	State
price	float	Price

Player Strategy:

Table: tbl_plyrstrat

Primary key:- strategy_id

Foreign key:- player_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
strategy_id	bigint	Strategy ID
player_id	varchar(50)	Player ID
Physical_strength	varchar(50)	Physical strength
speed	varchar(50)	Speed
mental_health	varchar(50)	Mental health
defending	varchar(50)	Defending ability
attacking	varchar(50)	Attacking ability
technical	varchar(50)	Technical ability

Sponsor Registration:

Table: tbl_sponsor

Primary key:- sponsor_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
sponsor_id	bigint	Sponsor ID
name	varchar(50)	name
address	varchar(250)	Address
mob	bigint	Mobile number
email	varchar(50)	Email address
team_name	varchar(50)	Team name
amount	float	Amount
valid_from	varchar(50)	Valid from
valid_to	varchar(50)	Valid to
status	varchar(50)	status

Stadium Management:

Table: tbl_stadium

Primary key:- stadium_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
stadium_id	int	Stadium ID
stadium_name	varchar(50)	Stadium name
capacity	bigint	capacity
place	varchar(50)	place

Match Fixtures :

Table: tbl_fixture1

Primary key:- fixture_id

FIELDS	DATATYPE/SIZE	DESCRIPTION
fixture_id	bigint	Fixture ID
home_team	varchar(50)	Home team
away_team	varchar(50)	Away team
stadium	varchar(50)	Stadium
date_m	varchar(50)	Date
TIME_M	varchar(50)	Time
result	varchar(50)	Results

5.5 PROCESS DESIGN

The different modules of the project are as follows::

- Login : Administrator should login to the system for the various processes
- Player Registration : This is used to register a new player to team and is done by admin.
- Player Strategy : Manage the details of various strategy of players, like attacking, defence and various other strategy.
- Player Transfer : Used to manage transfer of players from one team to another.
- Manage Team : Used to register a team.
- Team Strategy : Admin have the privilege to manage certain attributes of the Team Which are registered through manage team.
- Sponsors : The team need sponsors in order to contribute for their needs. This is used to register sponsors of teams in league.
- Fixtures : The match schedules or match between team ,time and venue are registered here.
- Stadium : The stadium of each match are managed here.

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. An elaborate testing of the data is prepared and the system is testing using this test data. While testing, errors are detected and corrected. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully in future. The testing steps are,

- Unit testing
- Integration testing
- Validation testing
- Alpha Testing
- Beta Testing

- **Integration testing**

Integration testing is a systematic technique for constructing tests to uncover errors associated with the interface. In this project, all the modules are combined and then entire program is tested as a while, thus in the integration testing step, all the errors uncovered for the next testing steps.

- **Validation testing**

Validation testing is where requirements stabled as a part of software requirement analysis is validated against the software that has been constructed. This test provides the final assurance that the software meet all functional, behavioural and performance requirements.

- **Alpha testing**

If the tested and real customers combining test the software in development site, then it is called alpha testing. Alpha testing is done at the developer's site by the testers. Alpha testing is the testing conducted by customers is developer's site. If the software is product, then beta testing will be done.

- **Beta testing**

Beta testing is the testing done by the customer's environment. Beta testing is the process of giving the product to customers and let them do the testing of their environment.

6.2 SYSTEM IMPLEMENTATION

The term implementation has different meanings, ranging from the conversion of basic application to a complete replacement of a computer system. The procedure, however, is virtually the same. Implementation is used here to mean the process of converting a new or a revised system, System design into an operational one. There are three types of implementations:

- 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

Here a manual department system is replaced by a computer system. Effective testing early in the process translated directly in to long-term cost is saving from a reduced number of errors. Back-up files are needed when the system is failure or down. The usability test verified the user-friendly nature of the system. Accurate and complete documentation is necessary for the user-friendly nature of the system.

System testing is designed to uncover weakness that is not found in the earlier tests. This includes forced system failure and validation of the total system, as its user in the operational environment will implement it. Generally, it begins with low volume of transactions base on live data. The volume is increased until the maximum level for each transaction type is reached. The total system is tested for recovery fall back after major failures to ensure that no data are lost during the emergency. All this is done with the old system still in operation after the candidate system passes the test, the old system is discontinued.

7. SECURITY TECHNOLOGIES AND POLICIES

7.1 SECURITY TECHNIQUES AND POLICIES

Maintenance involves the software industry captive, typing up system resources. It means restoring something to its original condition. Maintenance follows conversion to the extend. That change is necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files. Password protection and simple procedures to prevent the unauthorized access are provided to the users. The system allows the user to enter the system only through proper user name and password.

8. MAINTENANCE

8.1 MAINTENANCE

Structured design partitions a program into small, independent modules. They are arranged in a hierarchy that approximates a model of the business area and is organized in a top-down manner. Structured design is an attempt to minimize complexity and make a problem manageable by subdividing it into smaller segments, which is called modularization or decomposition. The primary advantage of this design is as follows:

- Critical interfaces are tested first.
- Early versions of the design, though incomplete, are useful enough to resemble the real system.
- Structuring the design provides control and improves morale.
- The procedural characteristics define the order that determines processing.

Modules that perform only one task are said to be less error-prone than modules that perform multiple tasks.

This part has the details regarding the project. It explains how this system works and what all procedures are taken to maintain the proper functioning of the new system. As the application is a web application, this start-stop-start nature of interaction of the web applications makes things a lot difficult. As our application is simpler any future enhancements can be fitted in this software. So it can run on any system.

9. CONCLUSION

9.1 CONCLUSION

The system objectives specified in the requirement documents are believed to have been satisfactory met. The system gives good result in the testing. The processing of the proposed system is simple and is in regular order. The various plans employed in the project are user friendly and easy to understand. The system has been designed in manner so as to provide valuable, timely and accurate information. The system was effective and efficient in the estimate generation process. Visual Basic 6.0 and MS SQL Server were the best selection for the system to be developed. Any native user can understand the problem, solve it and generate reports through the system.

.

10. BIBLIOGRAPHY

10.1 BIBLIOGRAPHY

BOOKS	AUTHORS
1. System Analysis and Design	Elias Mawad
2. An integrated Approach of Software Engineering	Pankaj Jalote
3. Guide to Visual Basic 6.0	Peter Norton
4. Introduction to Database Management	Naveen Prakash
5. Analysis and Design of Information System	V. Rajkumar

11. APPENDIX

11.1 CODES

Player Registration

```
Private Sub cboPreTeam_Click()

If rs.State = 1 Then rs.Close

rs.Open "select team_name from tbl_manageteam where team_id='" & cboPreTeam.Text &
"'", con

txtPreTeamName.Text = rs!team_name

End Sub

Private Sub cboTeamID_Click()

If rs.State = 1 Then rs.Close

rs.Open "select team_name from tbl_manageteam where team_id='" & cboTeamID.Text &
"'", con

rs.Open "select team_name from tbl_manageteam", con

Txtteamname.Text = rs!team_name

End Sub

Private Sub Cmdadd_Click()

If rs.State = 1 Then rs.Close

rs.Open "insert into tbl_plyrregis values ('" & TxtFirstName.Text & "','" &
TxtLastName.Text & "','" & DTPicker1.Value & "','" & cboTeamID.Text & "','" &
cboPreTeam.Text & "','" & TxtCountry.Text & "','" & TxtState.Text & "')", con,
adOpenDynamic, adLockBatchOptimistic

MsgBox "inserted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_plyrregis", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub


Private Sub Cmdclear_Click()
```

```
lbl_playerID.Caption = ""
TxtFirstName.Text = ""
TxtLastName.Text = ""
"DTPicker1.Value = ""
cboTeamID.Text = ""
cboPreTeam.Text = ""
TxtCountry.Text = ""
TxtState.Text = ""
End Sub
Private Sub cmdClose_Click()
Unload Me
End Sub
Private Sub Cmddelete_Click()
If rs.State = 1 Then rs.Close
rs.Open "delete from tbl_plyrregis where player_id='" & lbl_playerID.Caption & "'"
MsgBox "deleted"
If rs.State = 1 Then rs.Close
rs.Open "select * from tbl_plyrregis ", con, adOpenDynamic, adLockPessimistic
Set MSHFlexGrid1.DataSource = rs
MSHFlexGrid1.Visible = True
End Sub
Private Sub Cmdexit_Click()
Unload Me
End Sub
Private Sub CmdUpdate_Click()
res = MsgBox("Are you sure to update?", vbYesNo)
If res = vbYes Then
```

```
If rs.State = 1 Then rs.Close

rs.Open "update  tbl_plyrregis set first_name= '" & TxtFirstName.Text & "',last_name='" &
TxtLastName.Text & "',dob ='" & DTPicker1.Value & "', team_id ='" & cboTeamID.Text &
"',previous_team='" & cboPreTeam.Text & "',country='" & TxtCountry.Text & "', state='" &
TxtState.Text & "'   where player_id='" & lbl_playerID.Caption & "'", con, adOpenDynamic,
adLockOptimistic

MsgBox "Updated"

Form_Load

End If

End Sub

Private Sub CommandClear_Click()

lbl_playerID.Caption = ""

TxtFirstName.Text = ""

TxtLastName.Text = ""

"DTPicker1.Value = ""

cboTeamID.Text = ""

cboPreTeam.Text = ""

TxtCountry.Text = ""

TxtState.Text = ""

End Sub

Private Sub CommandDelete_Click()

If rs.State = 1 Then rs.Close

rs.Open "delete from tbl_plyrregis where player_id='" & lbl_playerID.Caption & "'"

MsgBox "deleted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_plyrregis ", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True
```

```
End Sub

Private Sub CommandSave_Click()

If rs.State = 1 Then rs.Close

rs.Open "insert into tbl_plyrregis values ('" & TxtFirstName.Text & "','" &
TxtLastName.Text & "','" & DTPicker1.Value & "','" & cboTeamID.Text & "','" &
cboPreTeam.Text & "','" & TxtCountry.Text & "','" & TxtState.Text & "'," & txtPrice.Text &
")", con, adOpenDynamic, adLockBatchOptimistic

MsgBox "inserted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_plyrregis", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub

Private Sub CommandUpdate_Click()

res = MsgBox("Are you sure to update?", vbYesNo)

If res = vbYes Then

If rs.State = 1 Then rs.Close

rs.Open "update  tbl_plyrregis set first_name= '" & TxtFirstName.Text & "',last_name='" &
TxtLastName.Text & "',dob ='" & DTPicker1.Value & "', team_id ='" & cboTeamID.Text &
"',previous_team='" & cboPreTeam.Text & "',country='" & TxtCountry.Text & "', state='" &
TxtState.Text & "',price=" & txtPrice.Text & "   where player_id='" & lbl_playerID.Caption
& "'", con, adOpenDynamic, adLockOptimistic

MsgBox "Updated"

Form_Load

End If

End Sub

Private Sub Form_Load()

getconnection

If rs.State = 1 Then rs.Close
```

```
rs.Open "select team_id from tbl_manageteam", con, adOpenDynamic,
adLockBatchOptimistic

While (rs.EOF = False)

cboTeamID.AddItem rs.Fields("team_id")

rs.MoveNext

Wend

If rs.State = 1 Then rs.Close

rs.Open "select team_id from tbl_manageteam", con, adOpenDynamic,
adLockBatchOptimistic

While (rs.EOF = False)

cboPreTeam.AddItem rs.Fields("team_id")

rs.MoveNext

Wend

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_plyrregis ", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub

Private Sub ImgStaffReport_Click()

If DataEnvironment1.Connection1.State = 1 Then DataEnvironment1.Connection1.Close

DataEnvironment1.Connection1.Open

DataEnvironment1.CommandPlayer lbl_playerID.Caption

DataReportPlayer.Show

End Sub

Private Sub MSHFlexGrid1_Click()

ImgStaffReport.Visible = True

lbl_playerID.Caption = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 1)

TxtFirstName.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 2)
```

```
TxtLastName.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 3)

DTPicker1.Value = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 4)

cboTeamID.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 5)

cboPreTeam.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 6)

TxtCountry.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 7)

TxtState.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 8)

txtPrice.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 9)

End Sub
```

Player Transfer

```
Private Sub CboNewTeamID_Change()

If rs.State = 1 Then rs.Close

 rs.Open "select team_name from tbl_manageteam where team_id='" &
CboNewTeamID.Text & "'", con

 lblNewTeam.Caption = rs!team_name

End Sub


Private Sub CboNewTeamID_Click()

If rs.State = 1 Then rs.Close

 rs.Open "select team_name from tbl_manageteam where team_id='" &
CboNewTeamID.Text & "'", con

 lblNewTeam.Caption = rs!team_name

End Sub


Private Sub CmdSearch_Click()

 If rs.State = 1 Then rs.Close

  '  rs.Open "select * from tbl_plyrregis where first_name like'" & txtSearch.Text & "%' or
last_name LIKE '" & txtSearch.Text & "%'"
```

```
    rs.Open "select * from tbl_plyrregis where first_name like'" & txtSearch.Text & "%' or
last_name LIKE '" & txtSearch.Text & "%' or player_id LIKE '" & txtSearch.Text & "%'",
conlblPlayerID.Caption = rs!player_id

  txtPlayerName.Text = rs!first_name

 LBL_PRE_TEAM_ID.Caption = rs!team_id

 End Sub


Private Sub Command1_Click()

res = MsgBox("Are you sure to Transfer?", vbYesNo)

If res = vbYes Then

If rs.State = 1 Then rs.Close

rs.Open "update  tbl_plyrregis set team_id= '" & CboNewTeamID.Text & "',price=" &
txtPrice.Text & "   where player_id='" & lblPlayerID.Caption & "'", con, adOpenDynamic,
adLockOptimistic

MsgBox "Updated"

Form_Load

End If

End Sub


Private Sub Form_Load()

getconnection

If rs.State = 1 Then rs.Close

rs.Open "select team_id from tbl_manageteam", con, adOpenDynamic,
adLockBatchOptimistic

While (rs.EOF = False)

CboNewTeamID.AddItem rs.Fields("team_id")

rs.MoveNext

Wend

End Sub
```

```
Private Sub LBL_PRE_TEAM_ID_Change()

If rs.State = 1 Then rs.Close

rs.Open "select team_name from tbl_manageteam where team_id='" &
LBL_PRE_TEAM_ID.Caption & "'", con

LBL_PRE_TEAM_NAME.Caption = rs!team_name

End Sub


Private Sub LBL_PRE_TEAM_ID_Click()


 If rs.State = 1 Then rs.Close

rs.Open "select team_name from tbl_manageteam where team_id='" &
LBL_PRE_TEAM_ID.Caption & "'", con

LBL_PRE_TEAM_NAME.Caption = rs!team_name

End Sub
```

Team Management

```
Private Sub Cmdadd_Click()

If rs.State = 1 Then rs.Close

rs.Open "insert into tbl_manageteam values ('" & Txtteamname.Text & "','" & Txtplace.Text
& "','" & Txtstatus.Text & "','" & Txtvalue.Text & "','" & cboStadium.Text & "','" &
Txtyearfound.Text & "')", con, adOpenDynamic, adLockBatchOptimistic

MsgBox "inserted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_manageteam", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub
```

```
Private Sub cmdClose_Click()

Unload Me

End Sub


Private Sub Cmddelete_Click()

If rs.State = 1 Then rs.Close

rs.Open "delete from tbl_manageteam where team_id='" & lblTeamID.Caption & "'"

MsgBox "deleted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_manageteam", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub


Private Sub Cmdexit_Click()

Unload Me

End Sub


Private Sub CmdUpdate_Click()

res = MsgBox("Are you sure to update?", vbYesNo)

If res = vbYes Then

If rs.State = 1 Then rs.Close

rs.Open "update  tbl_manageteam set team_name= '" & Txtteamname.Text & "',place='" &
Txtplace.Text & "',status ='" & Txtstatus.Text & "', value ='" & Txtvalue.Text &
"',stadium_name='" & cboStadium.Text & "',year_found='" & Txtyearfound.Text & "'", con,
adOpenDynamic, adLockOptimistic

MsgBox "Updated"

 Form_Load
```

```
End If

End Sub

Private Sub CommandDelete_Click()

If rs.State = 1 Then rs.Close

rs.Open "delete from tbl_manageteam where team_id='" & lblTeamID.Caption & "'"

MsgBox "deleted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_manageteam", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub


Private Sub CommandSave_Click()

If rs.State = 1 Then rs.Close

rs.Open "insert into tbl_manageteam values ('" & Txtteamname.Text & "','" & Txtplace.Text
& "','" & Txtstatus.Text & "','" & Txtvalue.Text & "','" & cboStadium.Text & "','" &
Txtyearfound.Text & "')", con, adOpenDynamic, adLockBatchOptimistic

MsgBox "inserted"

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_manageteam", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub


Private Sub CommandUpdate_Click()

res = MsgBox("Are you sure to update?", vbYesNo)

If res = vbYes Then

If rs.State = 1 Then rs.Close
```

```
rs.Open "update  tbl_manageteam set team_name= '" & Txtteamname.Text & "',place='" &
Txtplace.Text & "',status ='" & Txtstatus.Text & "', value ='" & Txtvalue.Text &
"',stadium_name='" & cboStadium.Text & "',year_found='" & Txtyearfound.Text & "'", con,
adOpenDynamic, adLockOptimistic

MsgBox "Updated"

 Form_Load

End If

End Sub


Private Sub Form_Load()

getconnection

If rs.State = 1 Then rs.Close

rs.Open "select stadium_name from tbl_stadium ", con

While (rs.EOF = False)

cboStadium.AddItem rs.Fields("stadium_name")

rs.MoveNext

Wend

If rs.State = 1 Then rs.Close

rs.Open "select * from tbl_manageteam", con, adOpenDynamic, adLockPessimistic

Set MSHFlexGrid1.DataSource = rs

MSHFlexGrid1.Visible = True

End Sub


Private Sub ImgStaffReport_Click()

If DataEnvironment1.Connection1.State = 1 Then DataEnvironment1.Connection1.Close

DataEnvironment1.Connection1.Open

DataEnvironment1.CommandTeam lblTeamID.Caption

DataReportTeam.Show
```

```
End Sub

Private Sub MSHFlexGrid1_Click()

ImgStaffReport.Visible = True

lblTeamID.Caption = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 1)

Txtteamname.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 2)

Txtplace.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 3)

Txtstatus.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 4)

Txtvalue.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 5)

cboStadium.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 6)

Txtyearfound.Text = MSHFlexGrid1.TextMatrix(MSHFlexGrid1.Row, 7)

End Sub


Private Sub Txtteamname_KeyPress(KeyAscii As Integer)

If (KeyAscii > 59 And KeyAscii < 96) Or (KeyAscii < 123 And KeyAscii > 96) Or
(KeyAscii = 32) Or (KeyAscii = 46) Or (KeyAscii = 8) Then

Else

KeyAscii = 0

MsgBox "Enter a Valid Name", vbInformation, "Valid Name"

End If

End Sub


Private Sub Txtvalue_KeyPress(KeyAscii As Integer)

If KeyAscii < 48 Or KeyAscii > 57 Then

   If Not KeyAscii = 8 Then

      KeyAscii = 0

      MsgBox "Enter a Number", vbInformation, "Valid Number"

   Else
```

```
        KeyAscii = 8

    End If

End If

End Sub
```

11.2 SCREEN SHOTS

MDI Form

Search Engine

Stadium Management

Fixtures

MDI form Exit