**ACKNOWLEDGEMENT**

I Sincerely owe our gratitude to all people who helped and guided us in completing this mini-project work.

I thankful to **Dr. BANSILAL**, Principal, NIEIT, Mysore. For having supported us in academic endeavors.

I thankful to **Sri. CHANDRU** A S, Head of Department of Information Science and Engineering, NIEIT for providing us timely suggestion, encouragement and support to complete this min-project.

I would like to sincerely thank our mini-project guides Sri. CHANDRU A S and Smt. SHRUTHI B S, faculties in dept. of Information & Science and Engineering for Providing relevant information, valuable guidance and encouragement to complete this min-project.

I would also like to thank all our teaching and non-teaching staff members of the Department, I grateful to the college for keeping labs open whenever required and providing us Systems and required software.

I always thankful to our parents for their valuable support and guidance in every step. Also thank all our friends for their support and guidance throughout the Mini-project.

I express our deepest gratitude and indebted thanks to NIEIT which has provided us an opportunity in fulfilling our most cherished desire of reaching our goal.

Yours Sincerely,

**PRANAV BHARATHI(4NN17IS015)**

**ABSTRACT**

I designed my mini-project using NetBeans (front end) and MYSQL(backend). It is very efficient in coding the mini-projects involved in the logical schema and hence its clean and clear to understand.

The NetBeans specification describes an abstract API provides access to components Managed by the service component runtime (Declarative Services). Provides support for running the Ant build tool in the platform. Provide support for performing structural and textual compare operations on arbitrary data and displaying results.

Football Tournament Management System project developed using NETBEANS JAVA and MYSQL.

This is a simple project, which helps to store database of TEAMS details, TEAMS performance and rankings.

I can use this project in college Inter Branch football tournaments, else in small FB Tournaments.

This type of System helps to crack the details of Teams performance in the Tournament. Hence, the system starts with login. Admin can edit, add, delete the details of Football Tournament, he has all type of authority.

**LIST OF CONTENTS**

**CHAPTERS PAGENO**

1. **Introduction 5**
   1. **About Database Management System 5**
   2. **Advantages of Database Management System 5**
   3. **Application of Database Management System 6**
   4. **Introduction to SQL 6**
   5. **DML (Data Manipulation Language) 7**
   6. **DDL (Data Definition Language) 8**
   7. **DCL (Data Control Language) 9**
   8. **TCL (Transaction Control Language) 9**
   9. **Introduction to Project 9**
   10. **About Football Tournament Management System 9**

**2. Requirement specification 10**

**2.1 Hardware requirement 10**

**2.2 Software requirement 10**

**3. SYSTEM DESIGN 12**

**3.1 E-R diagram 12**

**3.2 Logical schema 12**

**4. IMPLEMENTATION 15**

**4.1. Trigger Procedure 15**

**4.2. Stored Procedure 15**

**4.3. Modules 16**

**4.4. Table Description 16**

**4.5. Code Snippet 18**

**5. TESTING 21**

**5.1. Introduction to Testing 21**

**5.2. Stages in Implementation 21**

**5.3. Results 22**

**6. SNAPSHOTS 24**

**7. CONCLUSION 29**

**Future Enhancement 29**

**Bibliography 30**

**Chapter 1**

**INTRODUCTION**

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries reports, views, and other elements Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for ex) modeling the availability of rooms in hotels in a way that supports finding hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-user, others applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Ill known DBMS include Mysql, postgreSQL, EnterpriseDB, MongoDB, MariaDB, Microsft SQL server, oracle, Sybase,Sap , HANA, MemSQL, Sqlite and IBM DB2.

A database is not generally across different across DBMs, but different DBMs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support the most popular database systems since 1980s have all supported the relational model – generally associated with the SQL language. Sometimes a DBMs is loosely referred to as a “database”.

* 1. **About Database Management System**

In the 1970s and 1980s, attempts Ire made to build database systems with integrated hardware and software. The underlying philosophy was that such integration would provide higher performance at lower cost. Examples Ire IBM system/38, the early offering of tera data, database machine **C. Wayne Ratliff** the creator of dBASE stated : “ dBASE was different from programs like BASIC , C , FORTRAN ,and COBOL in that a lot of the dirty work had already been done . The data manipulation is done by DBASE instead of by the user can concentrate on what he is doing ,rather than having to mess with the dirty details of opening ,reading and closing files and managing space allocation ,” DBASE was one of the top selling software titles in the 1980s and early 1990s.

**1.2 Advantages of DBMS**

Databases are used to support internal operations of organization and to underpin online interactions with customers and suppliers.

Databases are used to hold administrative information and more specialized data , such as engineering data or economic models. Examples of database application include computerized library systems , flight reservation Systems ,computerized parts in inventory systems , and many content management systems that store websites as collections of webpages in a database.

* 1. **Application on Databases Management System**

A Databases Management System is a computerized record-keeping system. It is a repository or a container for collection of computerized data files. The overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contains in the database on demand. Information contained in the database on demand. Information can be anything that is of significance to an individual or organization.

Some of the following are the uses of DBMS:

1. Effective and efficient management of data
2. Query processing and management
3. Easy to understand and user friendly
4. Security and integrity of data
5. Data sharing and storage
   1. **Introduction to SQL**

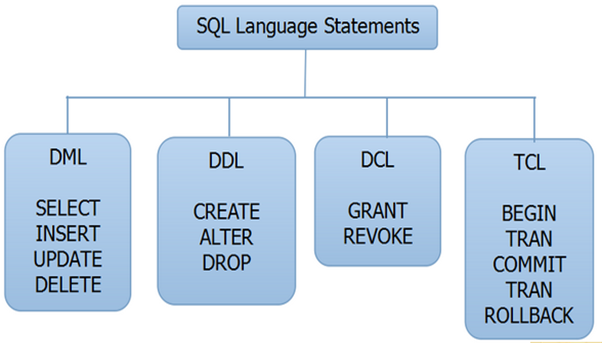
**SQL Structured query Language** is a domain- specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream processing in a relational data stream management system (RDBMS).

SQL devices in several ways from its theoretical foundation, the relational model and its tuple calculus. In that model , a table is a set of tuples, while in SQL ,tables and query results are list of rows the same may occur multiple times , and the order of rows can be employed in queries(e.g.. in the LIMIT clause).

Originally based upon relational algebra and tuple relational calculus, SQL consists of a **data definition language, data manipulation language,** and **data control language.** The scope of SQ includes data insert, query, update and delete, Schema creation and modification , and data access control .Although SQL is often described as, and to a great extent is , a declarative language ,it also includes procedural elements.

The four main categories of SQL statements are as follows:

1. DML (data Manipulation Language)
2. DDL (Data Definition Language)
3. DCL (Data Control Language)
4. TCL (Transaction Control Language)



* 1. **DML (Data Manipulation Language)**

DML statements affect records in table. These are basic operations I perform on data such as selecting a few records from a table, inserting new records .deleting unnecessary records , and updating/modifying existing records .

DML statements include the following:

**SELECT**– select records from a table

SYNTAX:

SELECT \* FROM <TABLENAME>

**INSERT**  - insert new records

SYNTAX:

INSERT INTO**<table name> values (value1, value2, value3…);**

**UPDATE**- update/modify existing records

SYNTAX:

Table\_ name SET column\_ name = value [column\_ name= value....] [WHERE condition]

**DELETE** - delete existing records.

SYNTAX:

DELETE FROM table\_ name [WHERE condition];

* 1. **DDL (Data Definition Language)**

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

**CREATE -** create a new Table, database, and schema

SYNTAX:

CREATE TABLE [table name] ([column definitions]) [table parameters];

**ALTER** – alter existing table, column description

SYNTAX:

ALTER object type, object name parameters;

**DROP -**delete existing objects from database

SYNTAX**:**

DROP Table<table name>;

* 1. **DCL (Data Control Language)**

DCL statements control the level of access that users have on database objects.

**GRANT** – allow the users to read /write on certain database objects.

**REVOKE**- keeps users from read/write permission on database objects.

* 1. **TCL (Transaction Control Language)**

TCL statements allow you to control and manage transactions to maintain the integrity of data within the SQL statements

**BEGIN** Transaction-opens a transaction

**COMMIT** Transaction-commits a Transaction

**ROLLBACK** Transaction- ROLLBACK a Transaction in case of any error

* 1. **Introduction to Project**

**Football tournament management system**

Football tournament management system is a system that will be used to manage football competition which using tournament system during the tournament. It will focus on managing all the important information during the tournament from registration process, during the tournament and until the tournament is over.

Currently most of the football organizations are filing method to store and manage all the information during the tournament period. The conventional method gives few complications for staff to handle all those data.

**Usefulness:**

Football tournament management system is developed to overcome the problems with

The current process and has its own objectives, which are:

1: To produce an efficient system which can manages players, coaches, teams and matches Information during the tournament.

2: To generate common reports that can be used during the football tournament such as league standing based on the match result.

**Chapter 2**

**REQUIREMENT SPECIFICATION**

A System requirements specification is a document or set of documentation that describes the features and behavior of a system of software application. It includes a variety elements that attempts to define the intended functionality required by the customer to satisfy their different uses. There are two types of requirements: Hardware and Software requirements.

In addition to specifying how the system should behave, the specification also defines at a high level the main business processes that will be supported, what simplifying assumptions have been made and what key performance parameters will need to be met by the system.

This document describes the nature of a project, software or application. This includes the purpose, scope, functional and nonfunctional requirements, software and hardware requirements of the project.

**2.1 Hardware requirements:**

It captures the complete hardware requirements for the system or a portion of the system. These requirements include the minimum processor speed, memory, and disc space required to install windows. In almost all cases, you will want to make sure that your hardware exceeds these requirements to provide adequate performance for the services and applications running on the server.

* Processor: Intel core processor
* RAM: 512 MB and above
* Hard-disc: 2 GB and above

**2.2 Software requirements:**

[Software requirements](https://en.wikipedia.org/wiki/Software_requirements) deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

A [computing platform](https://en.wikipedia.org/wiki/Computing_platform) describes some sort of [framework](https://en.wikipedia.org/wiki/Software_framework), either in [hardware](https://en.wikipedia.org/wiki/Computer_hardware) or [software](https://en.wikipedia.org/wiki/Software), which allows software to run. Typical platforms include a computer's [architecture](https://en.wikipedia.org/wiki/Computer_architecture), [operating](https://en.wikipedia.org/wiki/Operating_system) , or [programming languages](https://en.wikipedia.org/wiki/Programming_language) and their [runtime](https://en.wikipedia.org/wiki/Run-time_system) libraries.

[Operating system](https://en.wikipedia.org/wiki/Operating_system) is one of the requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained.

For example, most software designed for [Microsoft Windows XP](https://en.wikipedia.org/wiki/Microsoft_Windows_XP) does not run on [Microsoft Windows 98](https://en.wikipedia.org/wiki/Microsoft_Windows_98), although the converse is not always true. Similarly, software designed using newer features of [Linux](https://en.wikipedia.org/wiki/Linux) [Kernel](https://en.wikipedia.org/wiki/Linux_Kernel) v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

Software making extensive use of special hardware devices, like high-end [display adapters](https://en.wikipedia.org/wiki/Graphics_processing_unit), needs special [API](https://en.wikipedia.org/wiki/Application_programming_interface) or newer device drivers. A good example is [DirectX](https://en.wikipedia.org/wiki/DirectX), which is a collection of APIs for handling tasks related to multimedia, especially game programming, on [Microsoft](https://en.wikipedia.org/wiki/Microsoft) platforms.

* Operation system: Windows 10 Pro
* Front end: JAVA
* IDE: NETBEANS
* Back end: MySQL
* Programming language: SQL
* Server: XAMPP

**Chapter 3**

**System Design**

**INTRODUCTION:**

The system design document describes the system requirements, operating environment, system and sub system architecture, files and databases design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

**3.1 E-R DIAGRAM:**

An entity relationship (ER MODEL) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specify relationships that can exists between instances of those entity types. Fig 3.1 shows the ER diagram of Football Tournament management system.

**3.2 Logical Schema**

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among then are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagram. It’s the database designers who design to help programmers understand the database and make it useful. Fig 3.2 shows the Logical Schema of this project.

**Following shows the ER – Diagram of Project**

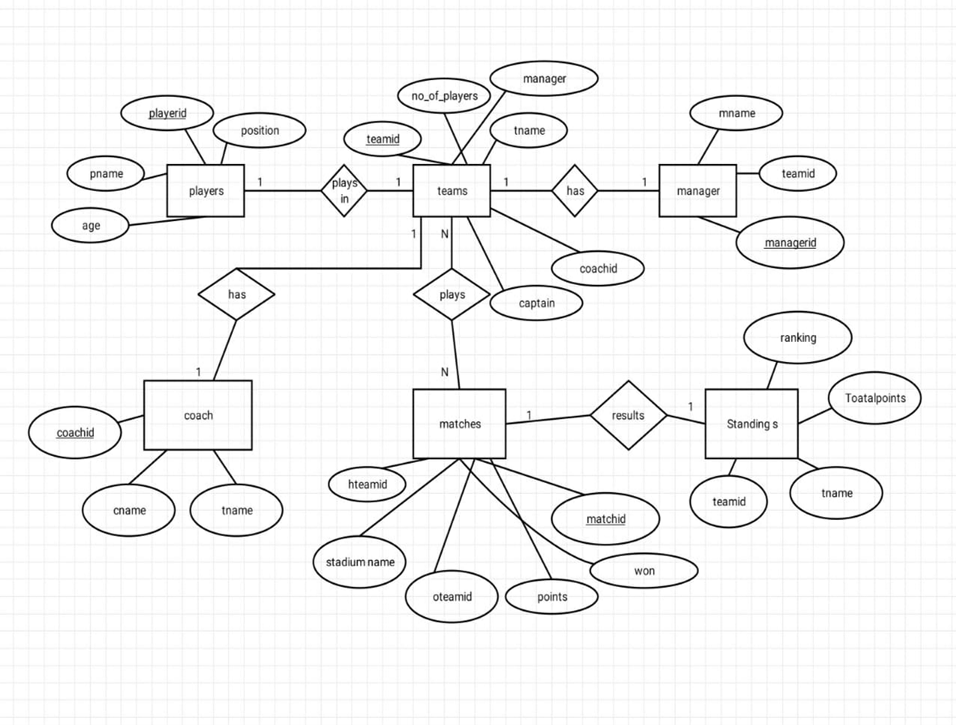
****

Fig 3.1: ER-Diagram of Football Tournament management system.

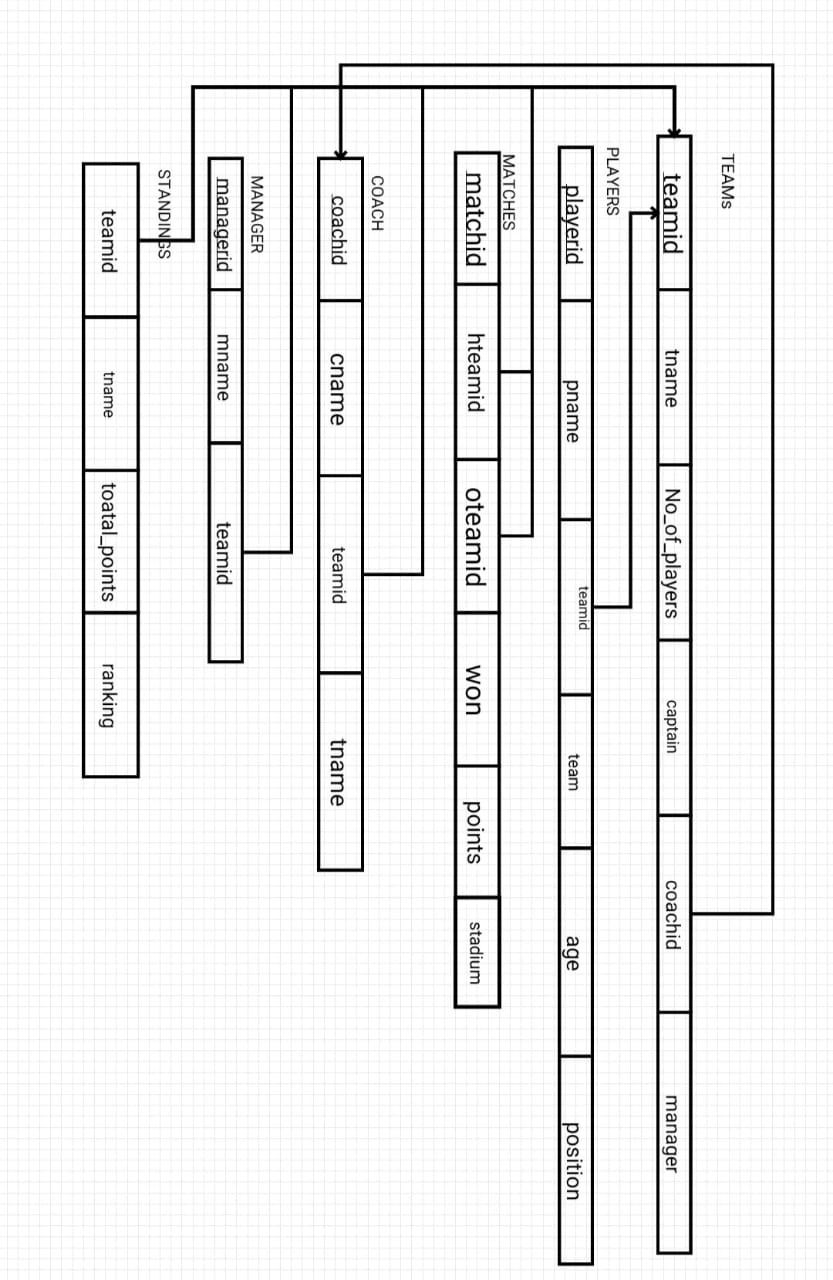


Fig 3.2: Logical Schema of Project

**Chapter 4**

**IMPLEMENTATION**

The special methods used in the project are explained:

**4.1. Triggers**

A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update.

A trigger is defined to activate when a statement inserts, update, or delete rows in the associated table. These row operations are trigger events. For example rows can be inserted by INSERT OR LOAD DATA statements, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

The Trigger used in the project is to store the duplicate values of the teams registration table. All the team details in the teams registration table are stored in the teamsbackup table. So I can view all the team details in the teamsbackup table also.

**Triggers Code Snippet**

DELIMITER $$

CREATE DEFINER=`root`@`localhost`TRIGGER`after\_teams\_insert`

AFTER INSERT ON `teams` FOR EACH ROW

BEGIN

INSERT INTO teams\_backup VALUES (NEW.teamid, NEW.tname,

NEW.total\_no\_of\_players, NEW.captain, NEW.coachid, NEW.manager\_name);

END

**4.2. Stored Procedure**

A procedure is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statement. All most all relational database system supports stored procedure, MYSQL 5 introduce stored procedure. In the following sections I have discussed MYSQL procedure in details and used MYSQL 5.6 under windows 7. MYSQL 5.6 supports routines and there are two kinds of routines: stored procedures which you call, or functions whose return values you use in other SQL statements the same way that you use pre-installed MYSQL functions like pi ().

The major difference is that UDFs can be used like any other expression within the SQL statements, whereas stored procedure must be invoked using a call state.

**Stored procedure Code Snippet**

DELIMITER $$

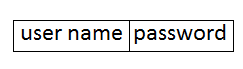
DROP PROCEDURE `getteams`;

CREATE DEFINER=`root`@`localhost`PROCEDURE`getteams`()

NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT \* FROM teams;

* 1. **Module**
* Admin
* Teams
* Players
* Matches
* Manager
* Standings
  1. **Table Description**

**Login**

****

The database provides volatile details such as Team details, Match details, Player details, Coach details, Manager details. It is highly necessary to protect these information from falling into wrong hands. Hence only certain Admin should be allow Id to access the database and in the system only the authorized users are allow Id to do so. The authorized users of the Football Tournament Management Database System can access the database by logging into the database, using their respective username and password. The usernames and passwords are also stored in the same database.

**Teams**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team id** | **T name** | **No\_of\_Players** | **captain** | **coachid** | **manager** |

Teams is the table used for storing complete information about the team. It is used to store details like Team id, T name, no\_of\_players, captain, coachid, manager.

**Players**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **playerid** | **pname** | **T name** | **teamid** | **age** | **position** |

Players is the table which contains the complete information about the player such as playerid, pname, age, position of the player in the team and of the player which he plays.

**Matches**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **matchid** | **Hteamid** | **oteamid** | **won** | **points** |

Matches is a table which contains the complete information of the match such as matched, hostteamid, opponentteamid, the team which win the match, points rewarded for the team and the stadium in which the match has been played.

**Coach**

|  |  |  |  |
| --- | --- | --- | --- |
| **coachid** | **Coach name** | **Team id** | **T name** |

It is the table which contains the information about the coach such as coachid, coachname, teamid and the team name for which he is appointed as a coach.

**Manager**

|  |  |  |
| --- | --- | --- |
| **Manager id** | **Manager name** | **T name** |

Manager table contains the information about managerid, managername and the team which he works as a manager. He is the one who manages the team details.

**Standings**

|  |  |  |  |
| --- | --- | --- | --- |
| **team id** | **T name** | **Total points** | **ranking** |

Standings table contains the information about the team performance in the tournament. Based on the results, Ranking is calculated.

* 1. **Code Snippet**

**FRONT End code in NetBeans JAVA for editing and updating the database**

Private void jbutton2action performed(java.awt.event.actionevent evt) {

// TODO add your handling code here:

Defaulttablemodel d1= (defaulttablemodel)jtable1.getmodel();

Int selectindex=jtable1.getselectedrow();

Try {

String id=d1.getvalueat(selectindex, 0).tostring();

String name=txtname.gettext();

String players= txtnoofplayers.gettext();

String caid= txtcaptainid.gettext();

String coid= txtcoachid.gettext();

String mname= txtmanagername.gettext();

Class.forname("com.mysql.jdbc.Driver");

Con=drivermanager.getconnection("jdbc:mysql://localhost/footballdb","root","");

Pst=con.preparestatement("update teams set tname=?,total\_no\_of\_players=?,captain=?,coachid=?,manager\_name=? Where teamid=?");

Pst.setstring(1, name);

Pst.setstring(2, players);

Pst.setstring(3, caid);

Pst.setstring(4, coid);

Pst.setstring(5,mname);

Pst.setstring(6,id);

Pst.executeupdate();

Joptionpane.showmessagedialog(this,"Record Updated......");

Table\_update();

} catch (classnotfoundexception ex) {

Logger.getlogger(teamsreg.class.getname()).log(Level.SEVERE, null, ex);

} catch (sqlexception ex) {

Logger.getlogger(teamsreg.class.getname()).log(Level.SEVERE, null, ex);

}

**FRONT End code in NetBeans JAVA for login page:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String USERNAME=txtusername.getText();

String PASSWORD=txtpass.getText();

if(USERNAME.equals("pranav")&& PASSWORD.equals("123")){

main m=new main();

this.hide();

m.setVisible(true);

}

else{

JOptionPane.showMessageDialog(this,"Username and password do not match");

}

}

**CHAPTER 5**

**TESTING**

**5.1 Introduction to testing**

Software testing is an investigation conducted to provide stake holders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test technique include the process of executing a program or application with the intent of finding software bugs (errors and defects) and verifying that the software product is fit for use.

**5.2 Stages in the implementation of testing**

* **Unit Testing**:

During this first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself.

The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Advantages such as improves design and allows better refactoring of code, reduces cost of testing as defects are captured in very early phase.

Techniques such as

black box testing- using which the user interface are tested,

white box testing-used to test each one of those functions behavior is tested,

gray box testing-used to execute tests.

All the entities i,e Teams registration, players, matches, coach, manager, Standings are working properly.

* **Integration testing:**

Integration testing allows individuals the opportunity to combine all the units within a program and test them as a group. This testing level is designed to find interfaces defects between the modules. This is beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they aren’t properly integrated, it will affect the functionality of the software program. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

All the user interfaces are integrated and working efficiently as per the requirements.

After integrating all the modules the system is working efficiently.

* **System testing:**

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets quality standards. System testing is undertaken by independent testers who haven’t played a role in developing the program. This testing is performed in an environment that closely mirrors production. System testing is very important because it verifies that the application meets the technical, functional and business requirements that Ire set by the customer*.*

* **Acceptance testing:**

The final level, acceptance testing is conducted to determine whether the system is ready for release. During the software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. Finally, all the connections in the entities are established efficiently in our project.

As per System requirements all the modules are working properly.

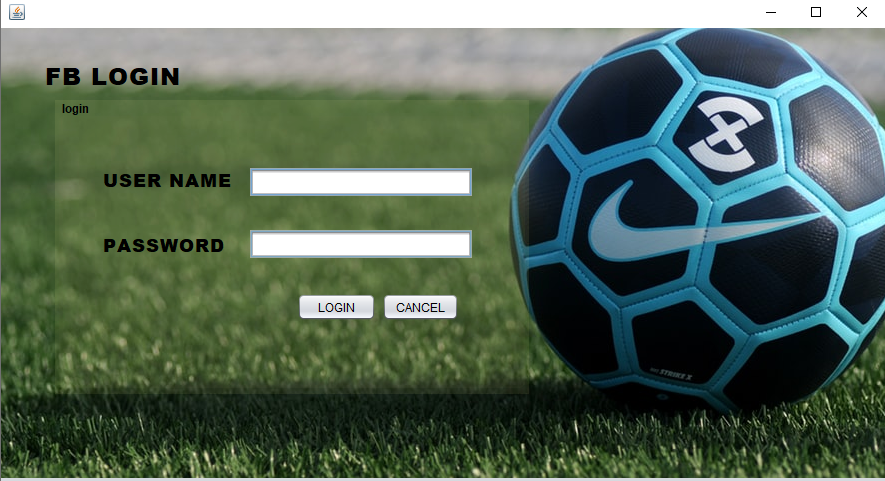
**5.3 Result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl.  no | Test case | Input Data | Steps to executed test case | Expected result | Pass / Fail |
| 1 | Correct username or password | After entering the data click the login button | After successful login, the Admin should enter the home page | The Admin enters the homepage | pass |
| 2 | Any of the 7 attribute buttons are clicked | Click on the teams registration button | The admin  should enter the  teams registration page | The admin  Enters the teams registration page | pass |
| 3 | Values are given to each field | Click on add, edit or delete buttons for execution | When clicked on add button, should add the data | A message is displayed that the data has been added | pass |
| 4 | Values are given to each field | Click on add, edit or delete buttons for execution | When clicked on delete button, should delete the data | A message is displayed that the data has been deleted | pass |
| 5 | Values are given to each field | Click on add, edit or delete buttons for execution | When clicked on edit button the values are should be edited | A message is displayed that the data has been updated | pass |

Several errors are detected and rectified and the whole project is working as it should with proper output and high efficiency.

**CHAPTER 6**

**SNAPSHOTS**



**Fig:6.1 Login page that reads Football Tournament Management System.**



**Fig:6.2 Home page depicting different modules in Football Tournament Management System.**



**Fig:6.3 Teams Registration details page.**

****

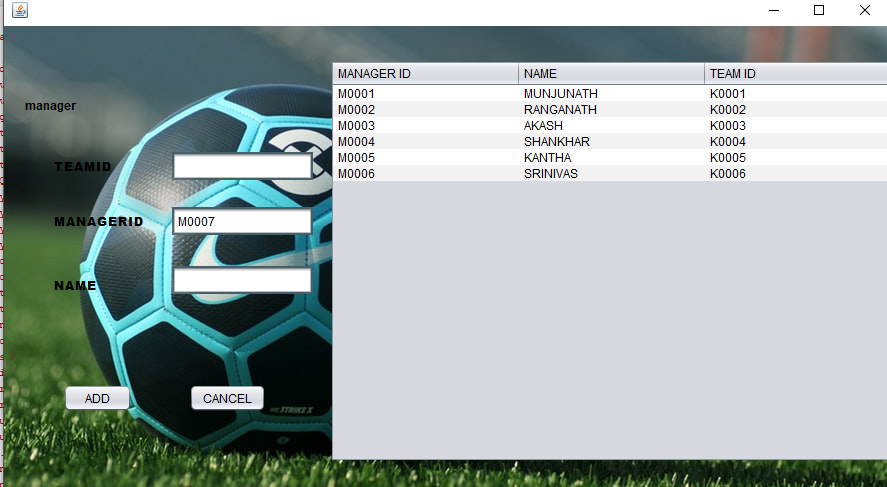
**Fig:6.4 Players details page**.



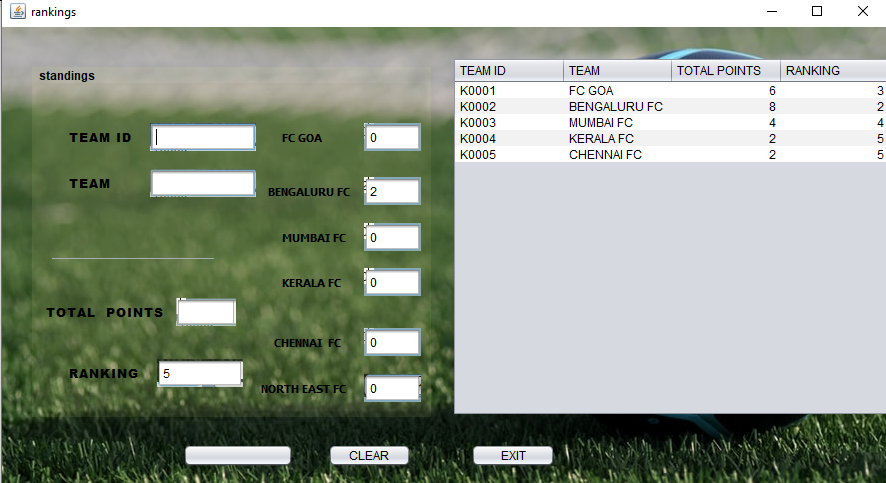
**Fig:6.5 Matches details page.**



**Fig:6.6 Coach details page**



**Fig:6.7 Manager details page.**



**Fig:6.6 Standings details page.**

**7.CONCLUSION**

Mine is a simple project, which helps to store the data base of teams details, Performance and Ranking.

I can use this project in college Inter Branch Football Tournaments or else in small Football Tournaments.

**FUTURE ENHANCEMENT**

The future Enhancement can be made include.

* If I wish to print a details of Teams, I can implement it.
* The admin interface can also be further embellished.

* In future I will allow for online reservations for teams.
* In future I providing the feedback and Support system.

**BIBILOGRAPHY**

Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.

Database Management Systems , Ramakrishna And Gerkhe , 3rd Edition ,2014 ,Mc Graw Hill.

Database systems models, languages Design and application Programming, Ramez Elmasri and Shamakant B .Navarthe , 7th edition, 2017, Pearson.

**websites**

[www.wikipedia.com](http://www.wikipedia.com)

[www.Stackoverflow.com](http://www.Stackoverflow.com)

[www.youtube.com](http://www.youtube.com)

[www.programmingknowledge.com](http://www.programmingknowledge.com)

[www.google.com](http://www.google.com)