# ELITE FOOTBALL CLUB MANAGEMENT SYSTEM

*Project Report Submitted By*

**SONA P VINOY**

**Reg. No: AJC17MCA-I052**

*In Partial fulfillment for the Award of the Degree Of*

**INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)**
**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

**AMAL JYOTHI COLLEGE OF ENGINEERING**
**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala–686518]

**2017-2022**

# DEPARTMENT OF COMPUTER APPLICATIONS

## AMAL JYOTHI COLLEGE OF ENGINEERING

## KANJIRAPPALLY

## CERTIFICATE

This is to certify that the Project report, "**ELITE FOOTBALL CLUB MANAGEMENT SYSTEM"** is the bonafide work of **SONA P VINOY (Reg.No:AJC17MCA-I052)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

**Mr. Binumon Joseph**
**Internal Guide**

**Ms. Meera Rose Mathew**
**Coordinator**

**Rev.Fr.Dr. Rubin Thottupurathu Jose**
**Head of the Department**

**External Examiner**

# DECLARATION

I hereby declare that the project report **“ELITE FOOTBALL CLUB MANAGEMENT SYSTEM”** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

**Date:** **SONA P VINOY**

**KANJIRAPPALLY** **Reg. No: AJC17MCA-I052**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms**. **Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Mr. Binumon Joseph** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

SONA P VINOY

# ABSTRACT

**Elite Football Club Management System** is about the management of a football league including various clubs. This project is a web application to manage football club that will enable football teams register its players, coaches and the matches they play. Since information management system has been of great concern to organizations, keeping records properly, and prompt retrieval is a key issue in record keeping. With the increasing number of clubs and players it is very difficult to manage the paper records when it comes to manual maintainence. But with proper football club management technology system, the club stands the chance of having up to date information about players and coaches including matches played and the result. It provides a convenient method of online registration for players and coaches. It also provides online ticket booking facility for the users according to the tournament events. Once the tournaments are scheduled the users can buy tickets by looking on seat availability and can also made the booking payment online. In this system there are mainly five users: Admin, Coach, Players and Users and an Organizer. New team registrations can be approved or rejected by admin after checking the details. After the team registration is successful admin can assign organizer for each team. Admin is the one who manages online ticket booking details. Users can check for the upcoming tournaments and can find the available seats for booking. Admin add update and delete teams and team head. Players are added to the particular team by each team head and can modify their information and position. Organizer is the one who assign coach for their corresponding team and also add team members. In this way team managament will be easy. Admin can add, update and delete new training programmes which can be viewed by the players and can enroll the programs. Also the player can apply leave which is approved or rejected by the admin. This system is made to help the customers for an easy and convenient way of ticket booking and the organization can manage the team and team members in an easy way.

# CONTENT

# List of Abbreviation

| | | |
|---|---|---|
| IDE | - | Integrated Development Environment |
| HTML | - | Hyper Text Markup Language. |
| CSS | - | Cascading Style Sheet |
| SQL | - | Structured Query Language |
| UML | - | Unified Modelling Language |

# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

**"ELITE FOOTBALL CLUB MANAGEMENT SYSTEM"** is a web application which is meant to help the football organization and the users. The customer can also reduce the time and effort in online ticket booking by using this system. Admin can easily manage the teams and schedule the events. The proposed system includes five users they are administrator, user, coach, player and organizer. Customers can login to the site and can make online ticket booking by looking into available seat categories through online. The approved players can enroll various training programmes which is generated by the admin and also can apply for leave which is approved or rejected by admin. The administrator has the central control over the whole system. The admin schedule the tournaments user can view the tickets. Team organizer is the one who add players to their respective teams and can make modifications and their corresponding position.

## 1.2 PROJECT SPECIFICATION

The proposed system includes four users they are The proposed system is made to help the customers for an easy and convenient way of booking the tickets online and also helps the organization to add update and delete new teams and their coaches and players. We will also provide users to give feedbacks, they can view the team and player details, payment details etc.

The system includes 5 modules. They are:

### 1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can approve or reject the new team registrations. Admin can view all the teams and player counts and tournament scheduling is done by admin and can manage the booking details.The leave applications from the players are managed by admin. And can view the leave history of each player. New training programmes are added and can view the enrolled players.

### 2. Customer Module

Customer can register and they can view the ticket for scheduled games and if interested they can register for a new team and wait for the approval. The ticket booking include online payment using razorpay gateway and can also print the booking details.

### 3. Organizer Module

Organizers are added by admin. Admin the is the one who assign organizer to each team. Organizer can view team details. Assign coach and players and their corresponding positions.

### 4. Player Module

Players can register in this site based on their qualification and then approved by the organization.Then each players are added to each team by the team head and give their position too. Each player is added to only one team ,duplication is not enabled.

### 5. Coach Module

Here the coach can see details including their team logo, team name, the category and the team description which can be viewed and managed.

# CHAPTER 2

# SYSTEM STUDY

## 2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analysed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## 2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Earlier all systems are handled manually in which the administrator has to maintain the details which leads to loss of data. Since the management system has been of great concern to organizations, keeping records properly, and prompt retrieval is a key issue in record keeping. Most often loss of vital documents are encountered, leading to poor management.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can easily get access to tournament schedules and appointment.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in managing the organization details..
- Project often delayed with no progress visibility.
- Human effort is needed.

## 2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for better growth; on such consideration the system is proposed. Proposed system is automated football league management system is a project which is aimed to develop an application in which admin manage various tournaments, games , scheduling , booking request , news and team and match results. Customer can register and log in to system, view the team, tournament schedule and request for ticket booking and online payment. This application also provide the functionality that the people can apply for both the players and coach selection. Mainly both the club and academy is integrated into a single application.

The team head can manage all the players in the respective teams and can add them to the desired position. The player details include their name, age, address, height and weight. The user can log into the system and can view all the team details and their player details. The scheduled tournaments can also be viewed by the customers and can make ticket booking online. The proposed system provides consistency of data and reduces difficulty in work.

Also, the admin can view the total teams and players in the organization and the ticket booking details are handled by the admin. This system helps the customer to done the ticket booking smoothly, efficiently and in less time and it helps the organization to increase the work. This system is made to help both the customer and the organization.

### 2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

○ **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining thedocuments.

○ **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

○ **Better service: -**

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

### 3.1.1 Economical Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving lot of time. The organization has in place, the required Hardware for implementing the football club proposed system. So the football club organization need not incur any additional expenditure.

The cost of project, ELITE FOOTBALL CLUB MANAGEMENT SYSTEM was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

### 3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of Technical Feasibility is defines as the feasibility that is concerned with specifying equipment and software that will successfully satisfy the user requirement. It compasses the technical needs of the system. It assesses the current resources such as hardware and software and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. It focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system.

According to feasibility analysis procedure the technical feasibility of the system is analysed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. It is evident that the necessary hardware and software are available for development and implementation of the football club proposed system.

We have used HTML,CSS,Javascript as front-end and php and MySQL as back-end and the software requirement is WAMP server. It is evident that necessary hardware and software are available for development and implementation of proposed system.

### 3.1.3 Behavioural Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

ELITE FOOTBALL CLUB MANAGEMENT SYSTEM, GUI is simple so that users can easily use it. ELITE FOOTBALL CLUB MANAGEMENT SYSTEM is simple enough so that no training is needed.

## 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

| | | |
|---|---|---|
| Processor | - | Intel core i3 |
| RAM | - | 4 GB |
| Hard disk | - | 1 TB |

### 3.2.2 Software Specification

| | | |
|---|---|---|
| Front End | - | HTML, CSS |
| Backend | - | MYSQL |
| Client on PC | - | Windows 7 and above. |
| Technologies used | - | JS, HTML5, AJAX, JQuery, PHP, CSS |

## 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP

PHP is a server side scripting language. It is used to develop static websites or dynamic websites or web applications that can interact with databases. PHP stands for Hypertext Pre-processor, that earlier stood for personal home Pages. PHP scripts can only be interpreted on a server that has PHP installed. It is a widely-used open source language that is specifically used for web application development and can be embedded within HTML. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page.

PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

### 3.3.2 MySQL

MySQL is an open source relational database management system. It runs as a server and allows multiple users to manage and create numerous databases. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database. SQL stands for "Structured Query Language" which is the standard language used to interact with databases.

- **MySQL is a database management system.**

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.

- **MySQL databases are relational.**

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.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

# CHAPTER 4

# SYSTEM DESIGN

## 4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. System design goes through two phases of development: Logical and Physical Design.

## 4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modelling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modelling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

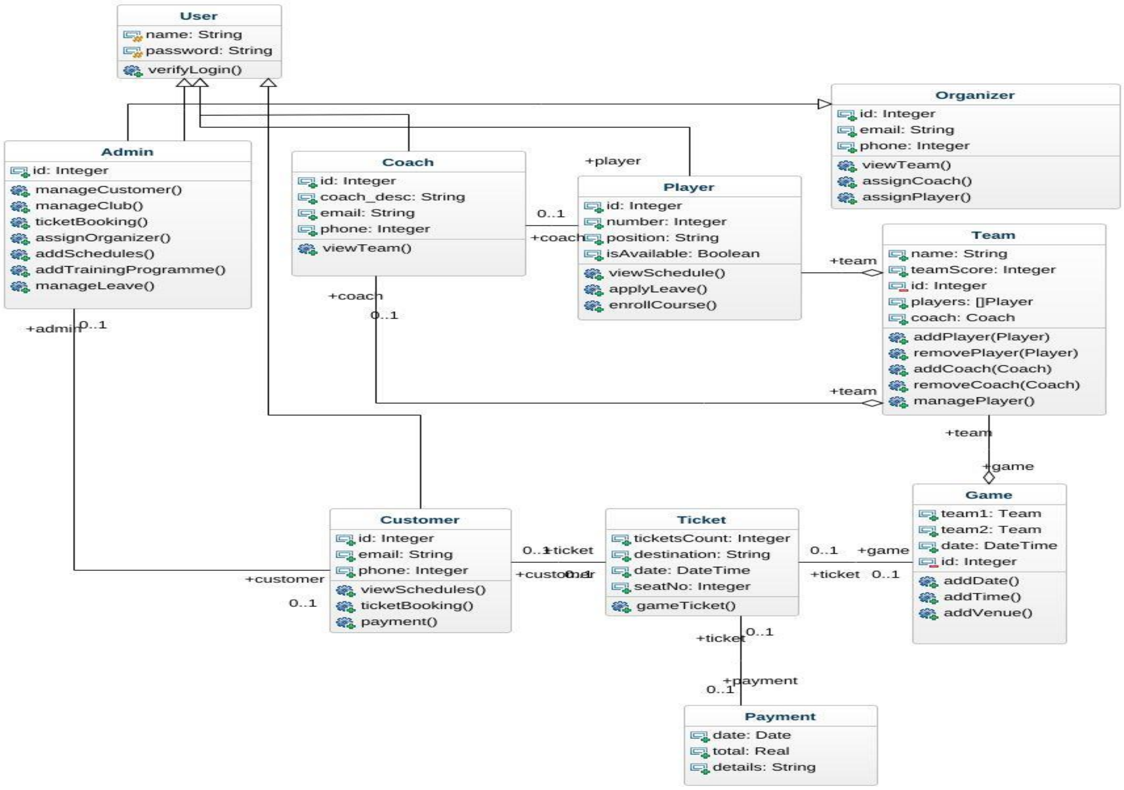
- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

## 4.2.1 CLASS DIAGRAM

A Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

**Figure 4.2.1 : Class Diagram for Elite Football Club**



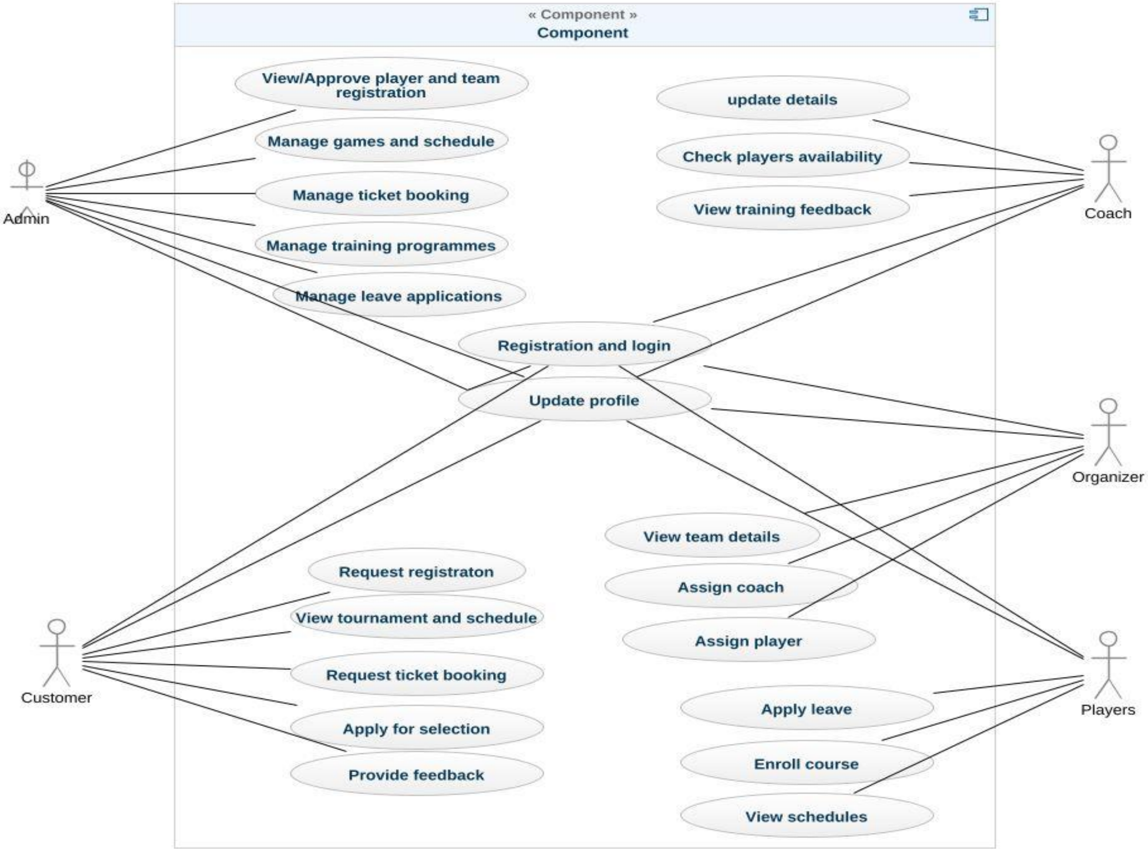
## 4.2.2 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems.

A use case diagram contains four components.

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

**Figure 4.2.2 : Use Case Diagram for Elite Football Club**



### 4.2.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

**Sequence Diagram Notations –**

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

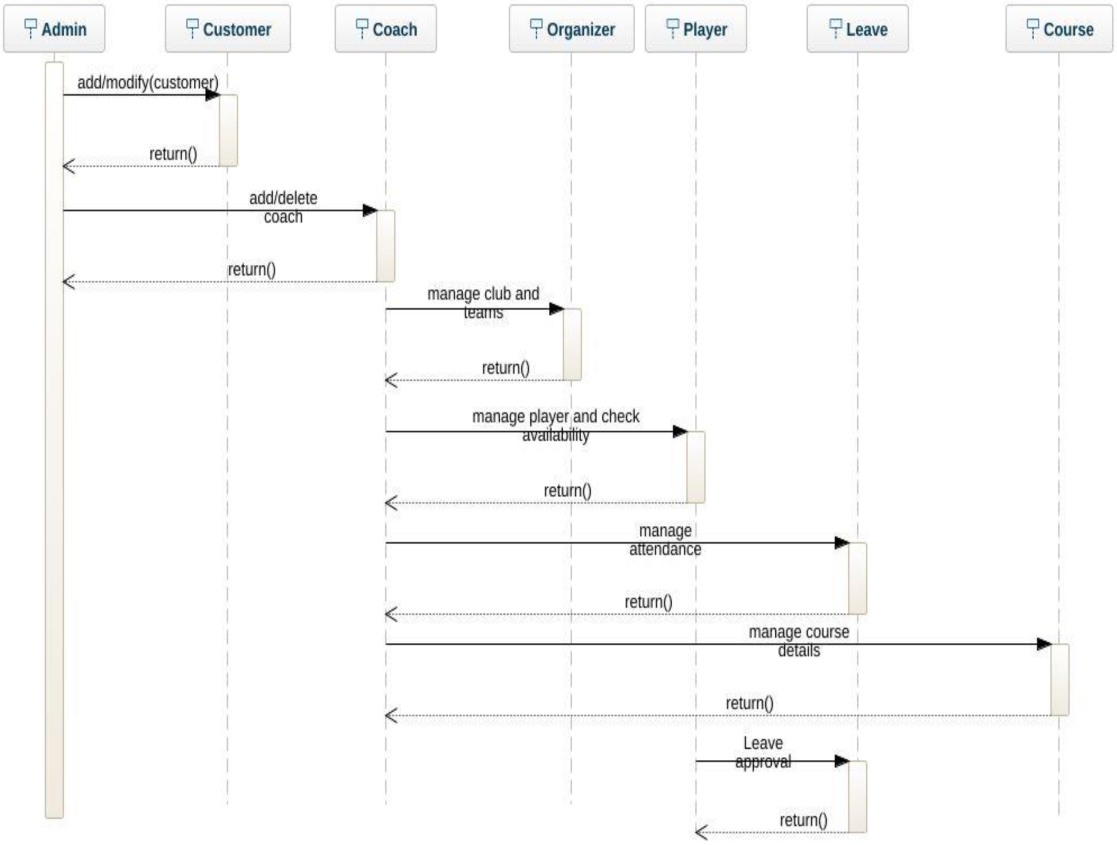
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

**iv. Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

**Uses of sequence diagrams –**

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

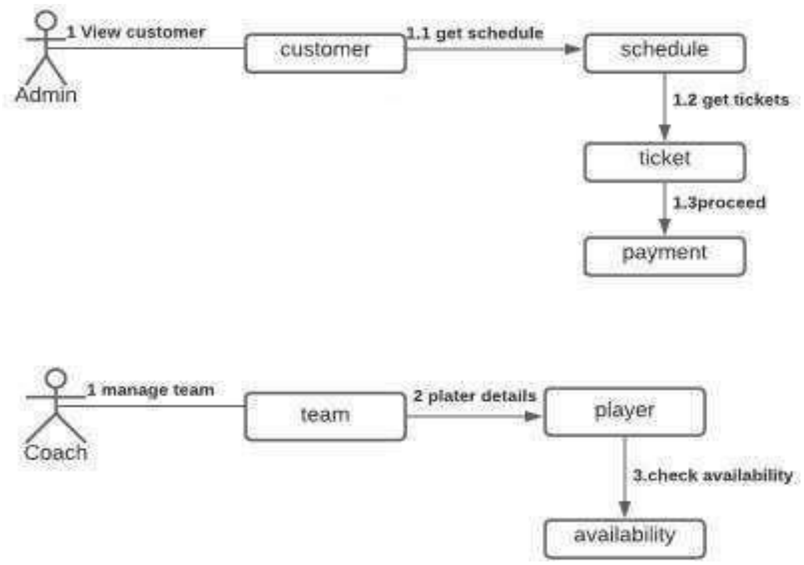
**Figure 4.2.3 : Sequence Diagram for Elite Football Club**



## 4.2.4 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

**Figure 4.2.4 : Collaboration Diagram for Elite Football Club**

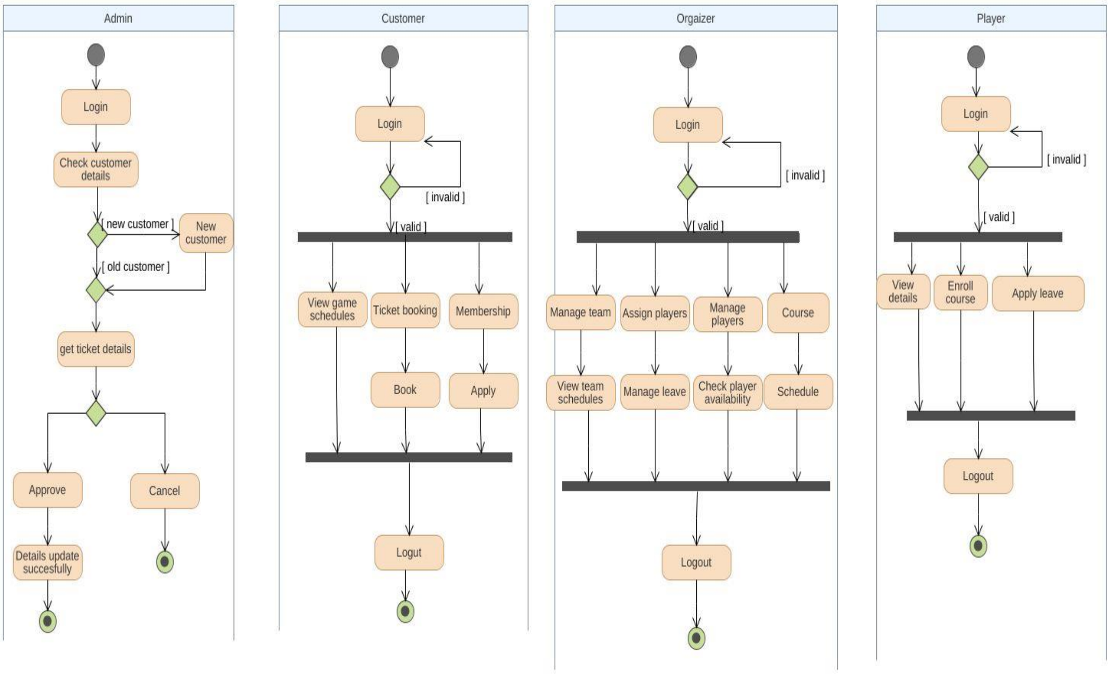


## 4.2.5 ACTIVITY DIAGRAM

An Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Figure 4.2.1 : Class Diagram for Elite Football Club

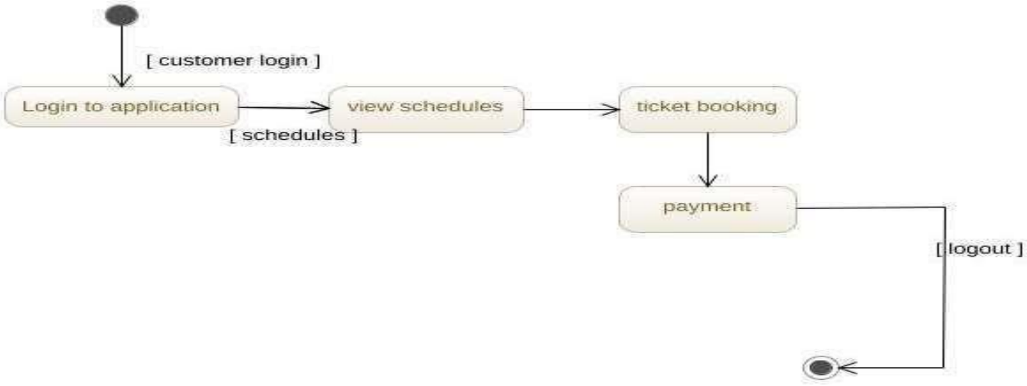


## 4.2.6 STATE CHART DIAGRAM

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.

Figure 4.2.6 : State Chart Diagram for Elite Football Club

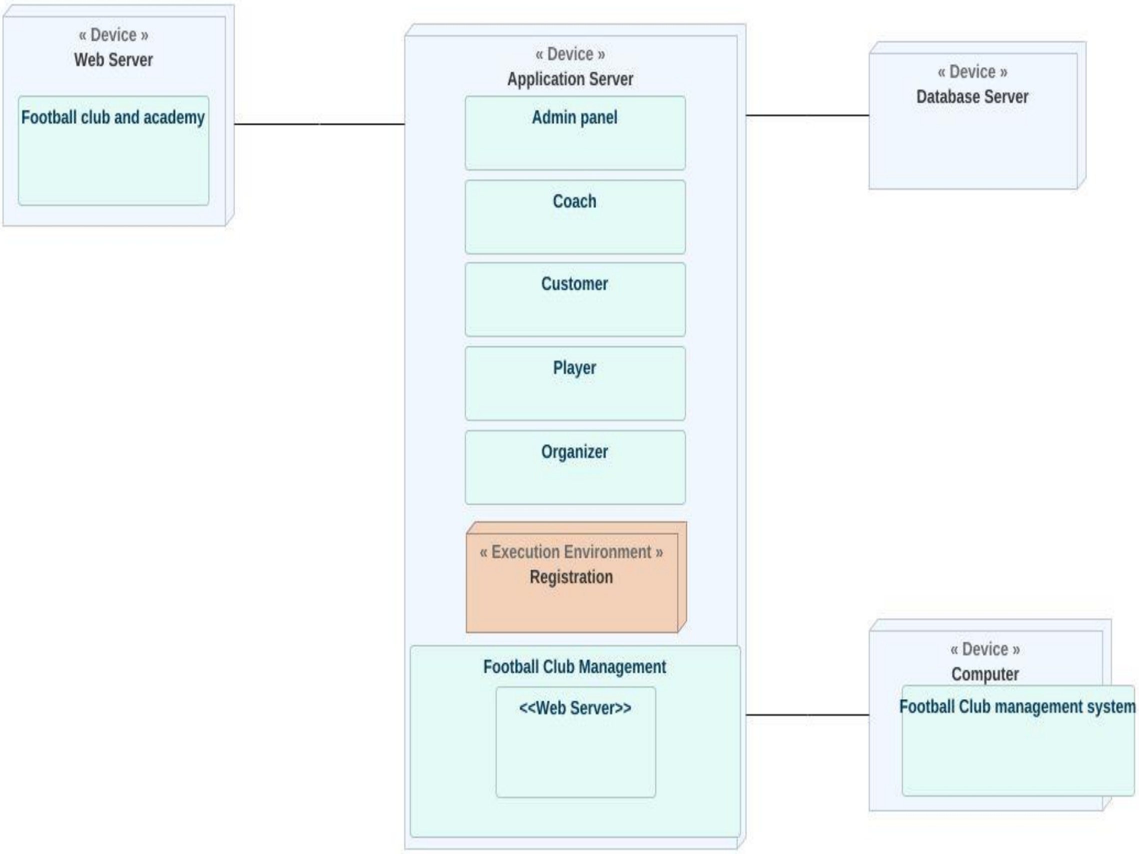


## 4.2.7 DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

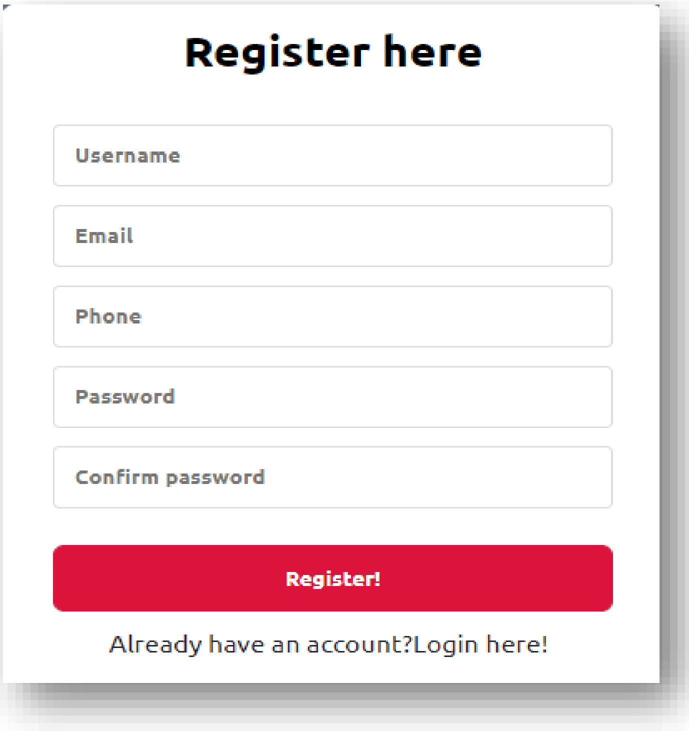
**Figure 4.2.7 : Deployment Diagram for Elite Football Club**



## 4.3 USER INTERFACE DESIGN USING FIGMA

### 4.3.1-INPUT DESIGN

**Form 4.3.1.1 : Form Name : Customer Registration**



Register here

Username

Email

Phone

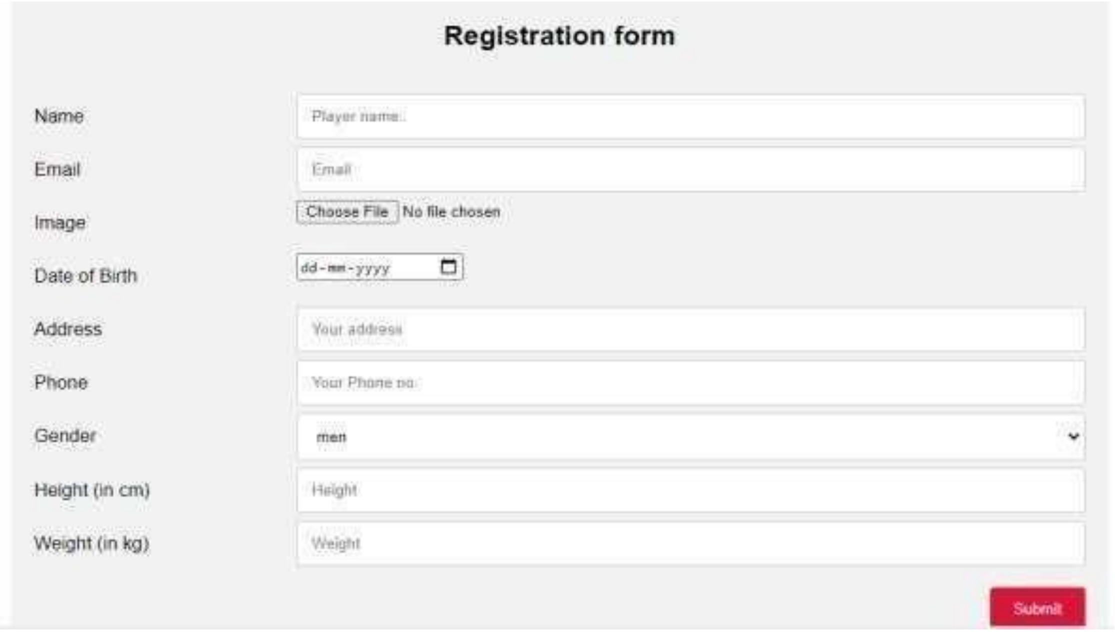
Password

Confirm password

Register!

Already have an account?Login here!

**Form 4.3.1.2 : Form Name : Player Registration**

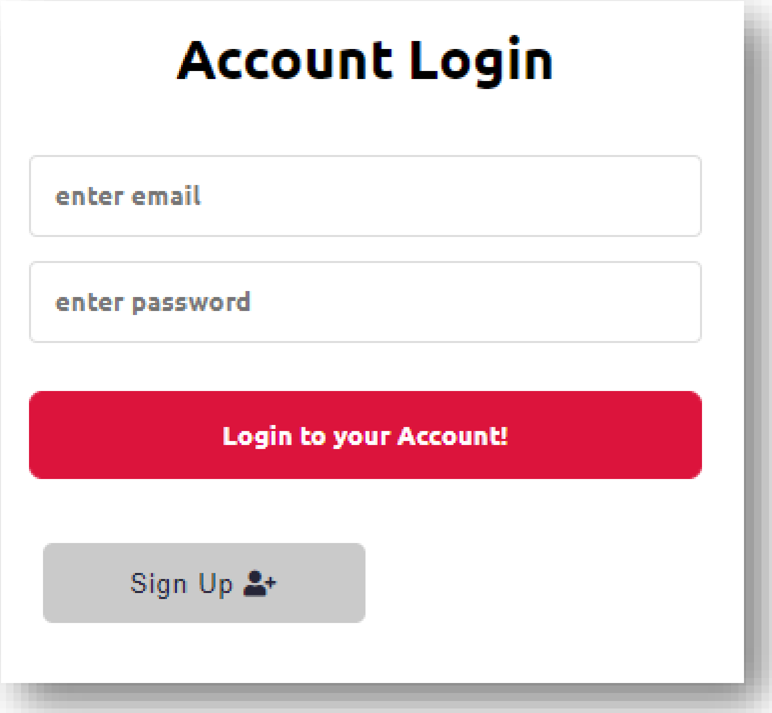


Registration form

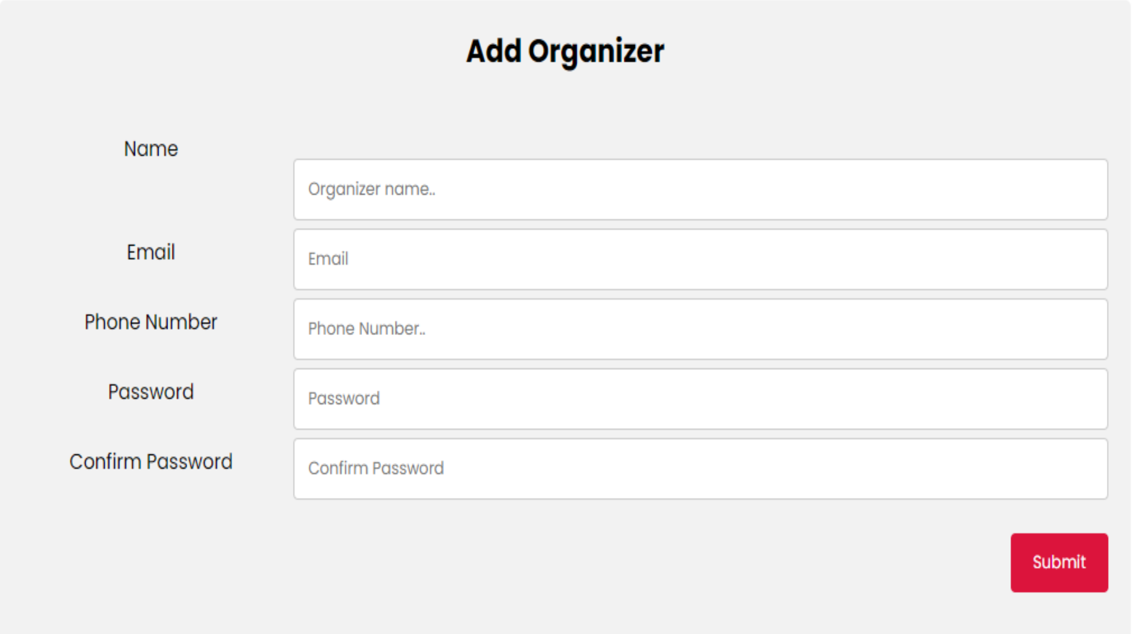
| | |
|---|---|
| Name | Player name |
| Email | Email |
| Image | Choose File No file chosen |
| Date of Birth | dd-mm-yyyy |
| Address | Your address |
| Phone | Your Phone no |
| Gender | men |
| Height (in cm) | Height |
| Weight (in kg) | Weight |

Submit

**Form 4.3.1.3 : Form Name : User Login**

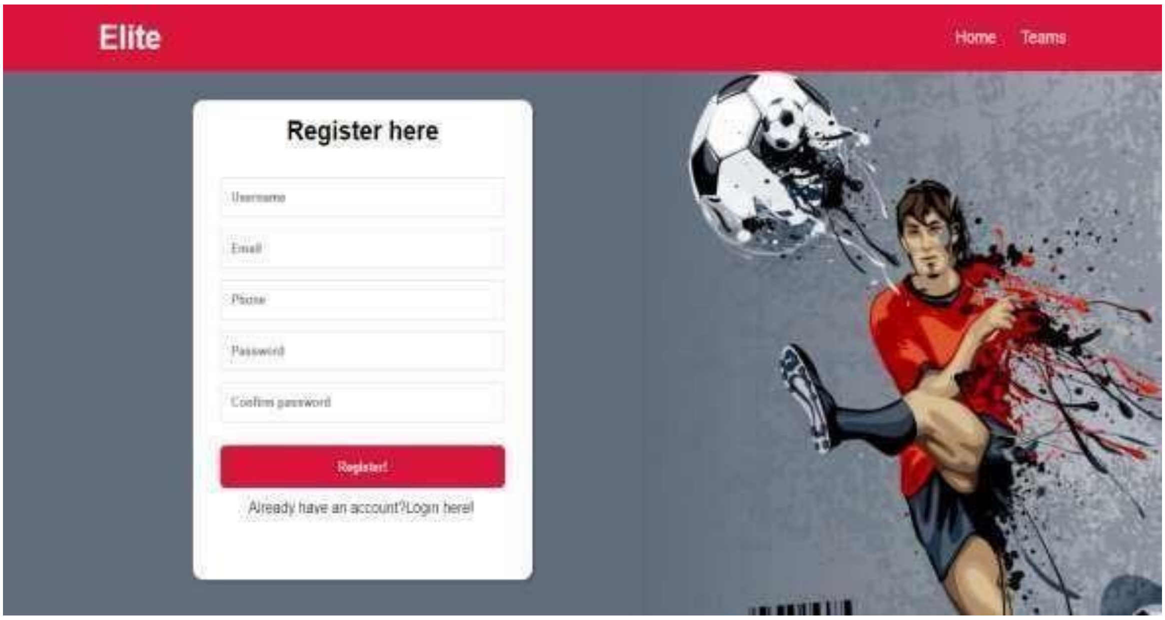


**Form 4.3.1.4 : Form Name : Add Organizer**

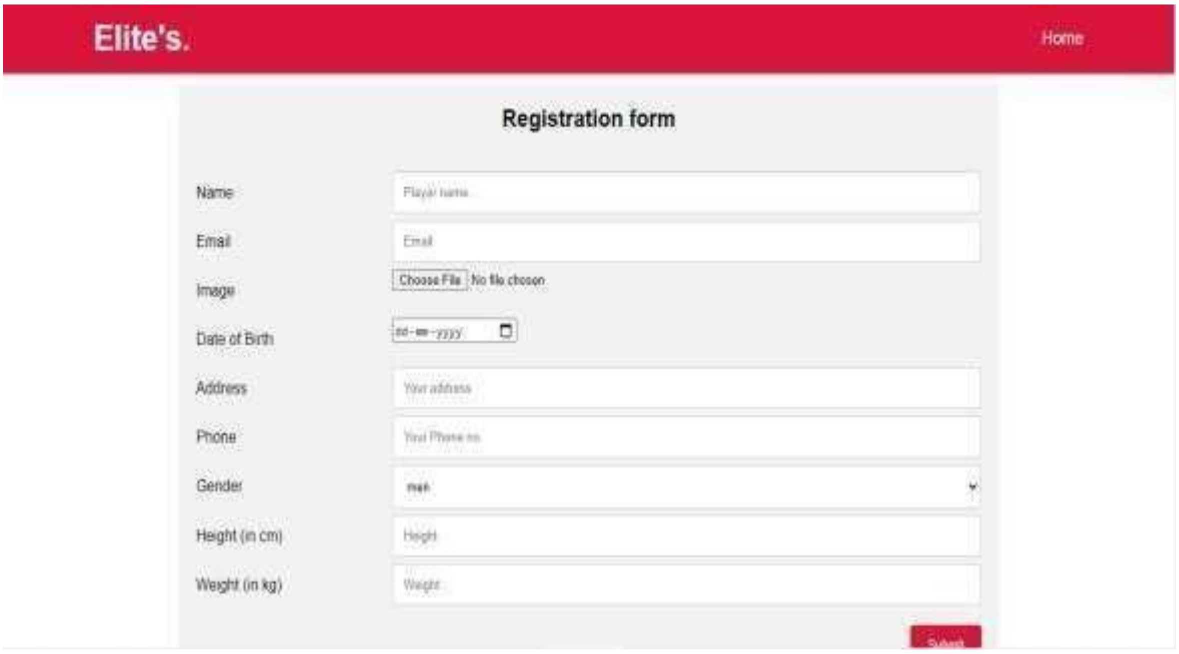


## 4.3.2 OUTPUT DESIGN

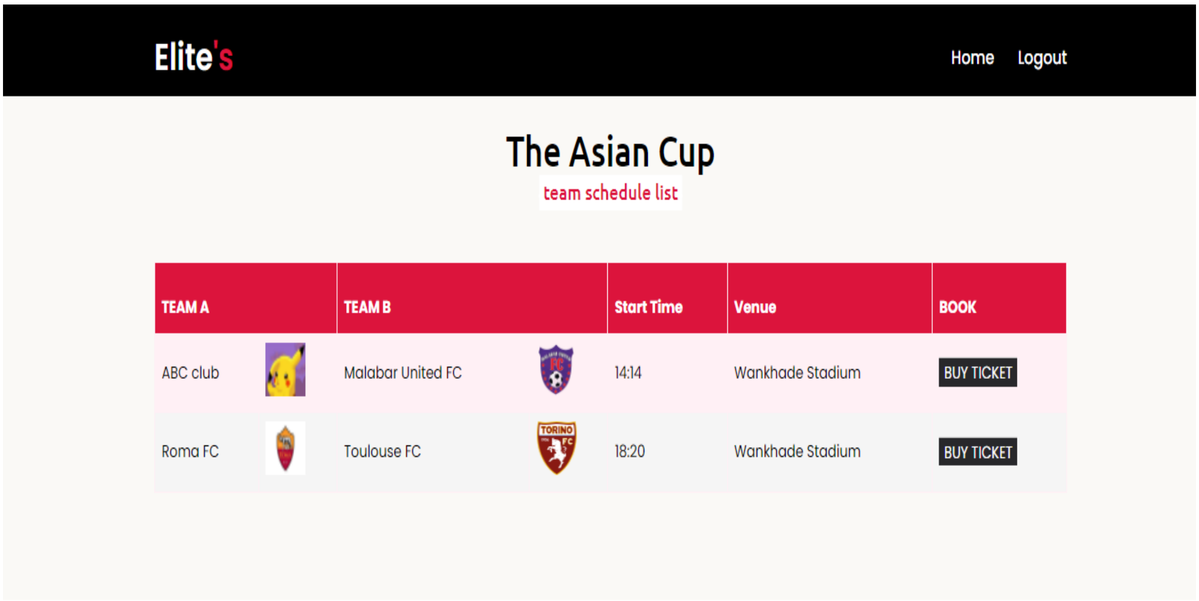
### Form 4.3.2.1 : Customer Registration



### Form 4.3.2.2 : Player Registration



## Form 4.3.2.3 : Customer Registration



The Asian Cup

team schedule list

| TEAM A | | TEAM B | | Start Time | Venue | BOOK |
|---|---|---|---|---|---|---|
| ABC club | | Malabar United FC | | 14:14 | Wankhade Stadium | BUY TICKET |
| Roma FC | | Toulouse FC | | 18:20 | Wankhade Stadium | BUY TICKET |

## 4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### 4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

**Relations, Domains & Attributes**

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

**Relationships**

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

### 4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from same table. As the name implies, it denotes putting things in the normal form.

The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

Normalize the data.

Choose proper names for the tables and columns.

Choose the proper name for the data.

**First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project.

**Second Normal Form**

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key

**Third Normal Form**

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

## 4.5 TABLE DESIGN

### Table 4.5.1 - tbl_register

**Table No : 01**

**Table Name : tbl_register**

**Primary Key : reg_id**

**Table Description : To store user Registration information**

| **Fieldname** | **Data Type** | **Size** | **Description** |
|---|---|---|---|
| **reg_id** | Int | 10 | Primary key of register table |
| **reg_name** | Varchar | 150 | Name of user |
| **reg_email** | Varchar | 150 | Email to login |
| **reg_phone** | Varchar | 50 | Phone Number of user |
| **reg_pwd** | Varchar | 150 | Password to login |
| **reg_status** | Int | 2 | Status of the user |
| **reg_date** | Timestamp | 6 | Date of registration |

### Table 4.5.2 - tbl_login

**Table No : 02**

**Table Name : tbl_ login**

**Primary Key : login_id**

**Foreign Key : reg_id, type_id**

**Table Description : To store login details**

| **Fieldname** | **Data type** | **Size** | **Description** |
|---|---|---|---|
| **login_id** | Int | 10 | Primary key |
| **reg_id** | Int | 10 | Register id of user |
| **type_id** | Int | 10 | Type id of user |

### Table 4.5.3 - tbl_ user_ type

**Table No** **: 03**

**Table Name** **: tbl_user_type**

**Primary Key** **: type_id**

**Table Description** **: To store type of user to login**

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **type_id** | Int | 10 | Primary key |
| **type_name** | Varchar | 150 | To store user type |

### Table 4.5.4 - tbl_ player_details

**Table No** **04**

**Table Name** **: tbl_player_details**

**Primary Key** **: p_id**

**Foreign Key** **: reg_id, gen_id**

**Table Description** **: To store player additional information**

| Fieldname | Date type | Size | Description |
|---|---|---|---|
| **p_id** | Int | 10 | Primary Key |
| **reg_id** | Int | 10 | Foreign key with reference from tbl_register |
| **gen_id** | Int | 10 | Foreign key with reference from tbl_gen_type |
| **p_image** | Varchar | 255 | To upload player photo |
| **p_dob** | Date | 20 | Date of Birth of player |
| **p_address** | Varchar | 255 | Address of player |
| **p_height** | Varchar | 100 | Height of player |
| **p_weight** | Varchar | 100 | Weight of player |
| **status** | Int | 10 | Status |

### Table 4.5.5 - tbl_gen_type

**Table No** : 05

**Table Name** : tbl_gen_type

**Primary Key** : gen_id

**Table Description** : To store gender type

| Fieldname | Data Type | Size | Description |
|---|---|---|---|
| **gen_id** | Int | 10 | Primary key |
| **gen_name** | Varchar | 100 | Gender type |

### Table 4.5.6 - tbl_ team

**Table No** : 06

**Table Name** : tbl_team

**Primary Key** : team_id

**Foreign Key** : gen_id

**Table Description** : To store team details

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **team_id** | Int | 10 | Primary key |
| **gen_id** | Int | 10 | Foreign key with reference from tbl_gen_type |
| **team_name** | Varchar | 150 | Team name |
| **team_logo** | Varchar | 225 | To store the team logo |
| **team_desc** | Varchar | 250 | To get the team description |
| **team_status** | Int | 2 | Status |
| **team_date** | Timestamp | 6 | Time of creation |

### Table 4.5.7 - tbl_ player_pos

**Table No** : 07

**Table Name** : tbl_player_pos

**Primary Key** : po_id

**Foreign key** : player_id, pos_id, team_id

**Table Description** : To store position team and player id

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **po_id** | Int | 10 | Primary key |
| **player_id** | Int | 10 | Foreign key with reference from tbl_register |
| **pos_id** | Int | 10 | Foreign key with reference from tbl_position |
| **team_id** | Int | 10 | Foreign key with reference from tbl_team |

### Table 4.5.8 - tbl_ team_coach_map

**Table No** : 08

**Table Name** : tbl_team_coach_map

**Primary Key** : map_id

**Foreign Key** : team_id, coach_id

**Table Description** : To store mapping details of team and coach

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **map_id** | Int | 10 | Primary key |
| **team_id** | Int | 10 | Foreign key with reference from tbl_team |
| **coach_id** | Int | 10 | Foreign key with reference from tbl_register |

## Table 4.5.9 - tbl_ team_organizer_map

**Table No** : 09

**Table Name** : tbl_team_organizer_map

**Primary Key** : to_id

**Foreign Key** : team_id, organizer_id

**Table Description** : To store mapping details of team and organizer

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **to_id** | Int | 10 | Primary key |
| **team_id** | Int | 10 | Foreign key with reference from tbl_team |
| **organizer_id** | Int | 10 | Foreign keywith reference from tbl_register |

## Table 4.5.10 - tbl_ team_organizer_map

**Table No** : 10

**Table Name** : tbl_tickets

**Primary Key** : ticket_id

**Foreign Key** :

**Table Description** : To store ticket category details

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **ticket_id** | Int | 10 | Primary key |
| **ticket_category** | Varchar | 250 | Display ticket categories |
| **tick_cat_desc** | Varchar | 250 | Category description |
| **tick_seats** | Int | 20 | Total seat count |
| **tick_price** | Int | 20 | Ticket price based on category |
| **tick_status** | Int | 2 | Status |

## Table 4.5.11 - tbl_ tournament

|  |  |
|---|---|
| **Table No** | **: 11** |
| **Table Name** | **: tbl_tournament** |
| **Primary Key** | **: trm_id** |
| **Foreign Key** | **:** |
| **Table Description** | **: To store tournament details** |

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **trm_id** | Int | 10 | Primary key |
| **trm_name** | Varchar | 150 | Tournament names |
| **trm_tot_teams** | Int | 10 | Total teams for tournament |

## Table 4.5.12 - tbl_ training_programme

|  |  |
|---|---|
| **Table No** | **: 12** |
| **Table Name** | **: tbl_training_programme** |
| **Primary Key** | **: tp_id** |
| **Foreign Key** | **:** |
| **Table Description** | **: To store training programme details** |

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **tp_id** | Int | 10 | Primary key |
| **tp_name** | Varchar | 255 | Programme name |
| **tp_start_date** | date |  | Programme start date |
| **tp_end_date** | date |  | Programme end date |
| **tp_desc** | Int | 20 | Training programme description |
| **status** | Int | 2 | Status |

### Table 4.5.13 - tbl_ course_enroll

**Table No** : 13

**Table Name** : tbl_course_enroll

**Primary Key** : ce_id

**Foreign Key** : course_id, reg_id

**Table Description** : To store course enroll details

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **ce_id** | Int | 10 | Primary key |
| **course_id** | Int | 10 | Foreign key with reference from tbl_training_programme |
| **reg_id** | Int | 10 | Foreign key with reference from tbl_register |

### Table 4.5.14 - tbl_leaves

**Table No** : 14

**Table Name** : tbl_leaves

**Primary Key** : id

**Foreign Key** : pid

**Table Description** : To store player leave application details

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **id** | Int | 10 | Primary key |
| **pid** | Int | 10 | Foreign key with reference from tbl_register |
| **pname** | Varchar | 255 | Player email |
| **descr** | Varchar | 255 | Reason and description |
| **fromdate** | date | | Leave from date |
| **todate** | date | | Leave end date |
| **status** | varchar | 255 | Leave status |

## Table 4.5.15 - tbl_ payment

**Table No** **: 15**

**Table Name** **: tbl_payment**

**Primary Key** **: book_id**

**Foreign Key** **: tv_d, tvs_id, user_id**

**Table Description** **: To store player leave application details**

| Fieldname | Data type | Size | Description |
|---|---|---|---|
| **book_id** | Int | 10 | Primary key |
| **tv_id** | Int | 10 | Foreign key with reference from tbl_booking |
| **tvs_id** | Int | 10 | Foreign key with reference from tbl_tournament |
| **user_id** | Int | 10 | Foreign key with reference from tbl_register |
| **no_seats** | Int | 10 | Total seats |
| **amount** | Int | 10 | Total amount |
| **status** | Int | 10 | Leave status |

# CHAPTER 5

# SYSTEM TESTING

## 5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are: Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include:

- Unit Testing
- Integration Testing
- Data Validation Testing
- Output Testing

### 5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

### 5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. Moreover differences in program structures were removed and a unique program structure was evolved.

### 5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

### 5.2.4 Automation Testing

The Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs.

### 5.2.5 Selenium Testing

Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation. It provides a playback tool for authoring functional tests without the need to learn a test scripting language. The Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high-quality web applications whether they are responsive, progressive, or regular.

**Table 5.2.5.1 : Test cases for a Login Page**

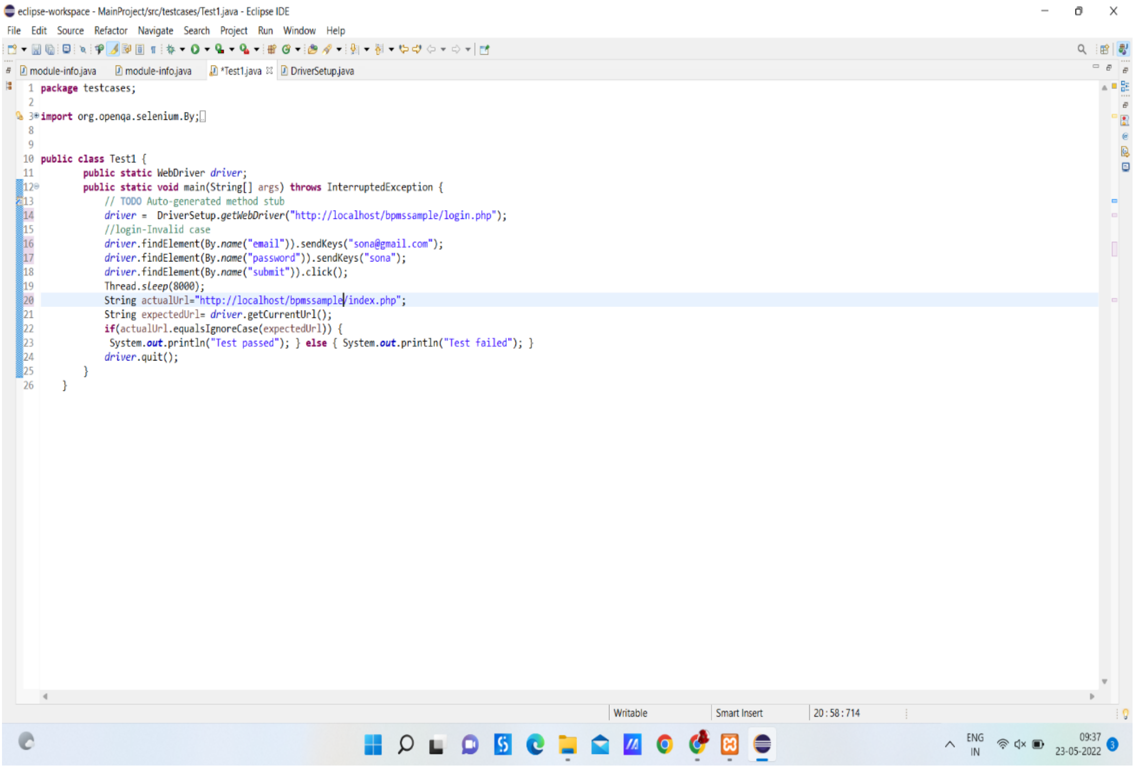
| | | | | | |
|---|---|---|---|---|---|
| **Project Name: Elite Football Club Management System** | | | | | |
| **Login Test Case** | | | | | |
| **Test Case ID:** Fun_1 | | | **Test Designed By: Sona P Vinoy** | | |
| **Test Priority (Low/Medium/High):** High | | | **Test Designed Date:** 17-05-2022 | | |
| **Module Name**: Login Screen | | | **Test Executed By: Binumon Joseph** | | |
| **Test Title:** Verify login with valid email and password | | | **Test Execution Date:** 19-05-2022 | | |
| **Description:** Test the Login Page | | | | | |
| **Pre-Condition: User** has valid email id and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigation to Login Page | | Login Page should be displayed | Login page displayed | Pass |
| 2 | Provide Valid Email Id | User Name: sona@gmail.com | User should be able to Login | User Logged inand navigated to Subadmin Dashboard with records | Pass |
| 3 | Provide Valid Password | Password: sona | | | |
| 4 | Click on Sign In button | | | | |
| 5 | Provide Invalid Email Id or password | Email Id: user@gmail. Com<br>Password: User12345 | User should not be able to Login | Message for enter valid email id or password displayed | Pass |
| 6 | Provide Null Email Id or Password | Email Id: null<br>Password: null | | | |
| 7 | Click on Sign In button | | | | |

**Post-Condition:** User is validated with database and successfully login into account. The Account session details are logged in database

## Code

```
package testcase;
import org.openqa.selenium.By;
import
org.openqa.selenium.WebDriver;
import
browserimplement.DriverSetup;
public class Firsts {
public static WebDriver driver;
public static void main(String[] args) throws InterruptedException {
// TODO Auto-generated method stub
driver =  DriverSetup.getWebDriver("http://localhost/bpmssample /login.php");
//login-Invalid case
driver.findElement(By.name("email")).sendKeys("sona@gmail.com");
driver.findElement(By.name("psw")).sendKeys("sona");
driver.findElement(By.name("submit")).click();
Thread.sleep(8000);
String actualUrl="http://localhost/bpmssample/index.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); } else { System.out.println("Test
failed");
}
driver.quit();
}
}
```

## Screenshot 5.2.5.1 : Testing 1 Code

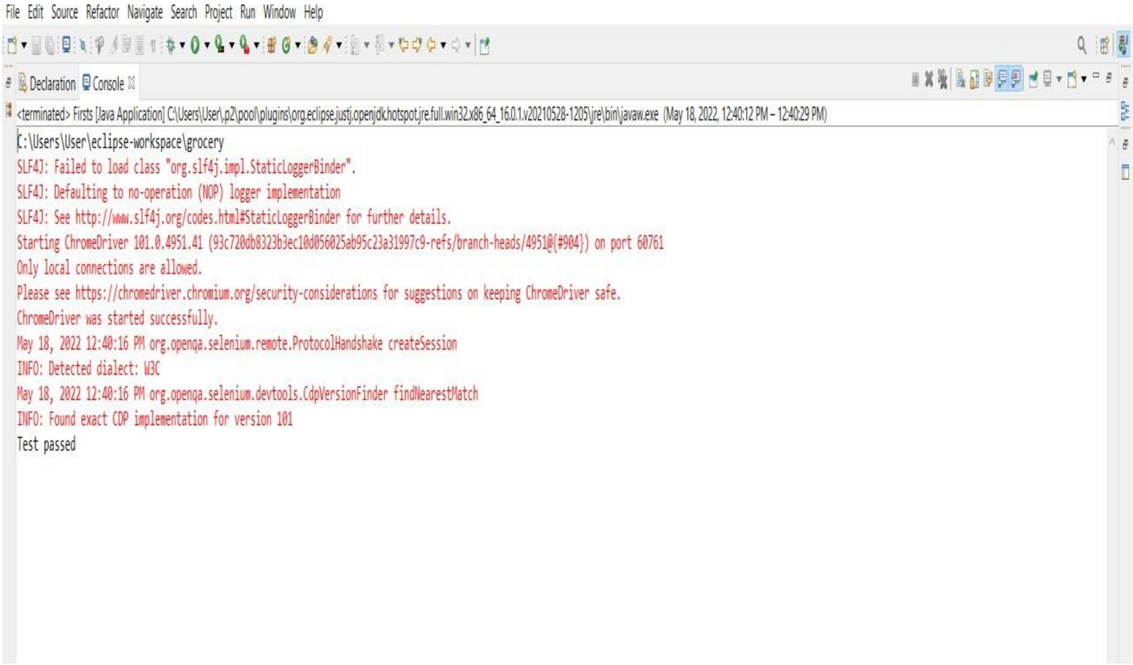

eclipse-workspace - MainProject/src/testcases/Test1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
module-info.java
module-info.java
*Test1.java
DriverSetup.java
Writable
Smart Insert
20 : 58 : 714
ENG IN
09:37
23-05-2022

```java
package testcases;

import org.openqa.selenium.By;


public class Test1 {
        public static WebDriver driver;
        public static void main(String[] args) throws InterruptedException {
            // TODO Auto-generated method stub
            driver =  DriverSetup.getWebDriver("http://localhost/bpmssample/login.php");
            //login-Invalid case
            driver.findElement(By.name("email")).sendKeys("sona@gmail.com");
            driver.findElement(By.name("password")).sendKeys("sona");
            driver.findElement(By.name("submit")).click();
            Thread.sleep(8000);
            String actualUrl="http://localhost/bpmssample/index.php";
            String expectedUrl= driver.getCurrentUrl();
            if(actualUrl.equalsIgnoreCase(expectedUrl)) {
             System.out.println("Test passed"); } else { System.out.println("Test failed"); }
            driver.quit();
        }
    }
```

## Screenshot 5.2.5.2 : Testing Output


File Edit Source Refactor Navigate Search Project Run Window Help
Declaration
Console

```
<terminated> Firsts [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (May 18, 2022, 12:40:12 PM – 12:40:29 PM)
C:\Users\User\eclipse-workspace\grocery
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

**Test Case for Edit Player Details Page**

| | |
|---|---|
| **Project Name: Elite Football Club Management System** | |
| **Update Test Case** | |
| **Test Case ID:**editPlayerDetails | **Test Designed By: Sona P Vinoy** |
| **Test Priority(Low/Medium/High):**High | **Test Designed Date:** 17-05-2022 |
| **Module Name**:Login Screen | |
| **Test Title :** Update player details | **Test Execution Date:** 19-05-2022 |
| **Description:** Login to system and update player details, if some error occurs, test will fail | |
| **Pre-Condition :**User has valid email and password | |

| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass /Fail)** |
|---|---|---|---|---|---|
| 1 | Navigation to Login Page | | Login Page should be | Login page displayed | Pass |
| 2 | Provide Valid Email | User Name: sona@gmail.com | User should be able to Login | User Logged in and navigated to Player details with records | Pass |
| 3 | Provide Valid Password | Password: sona | | | |
| 4 | Click on Login button | | | | |
| 5 | Provide player information | Input player details | User will be redirected to info page | User will be redirected to info page | Pass |
| 7 | Click on Edit button | | | | |
| 8 | Provide invalid informations | Input invalid house details | User will be redirected to info page | User will be stay on that page showing error message | Pass |
| 9 | Click on Edit button | | | | |

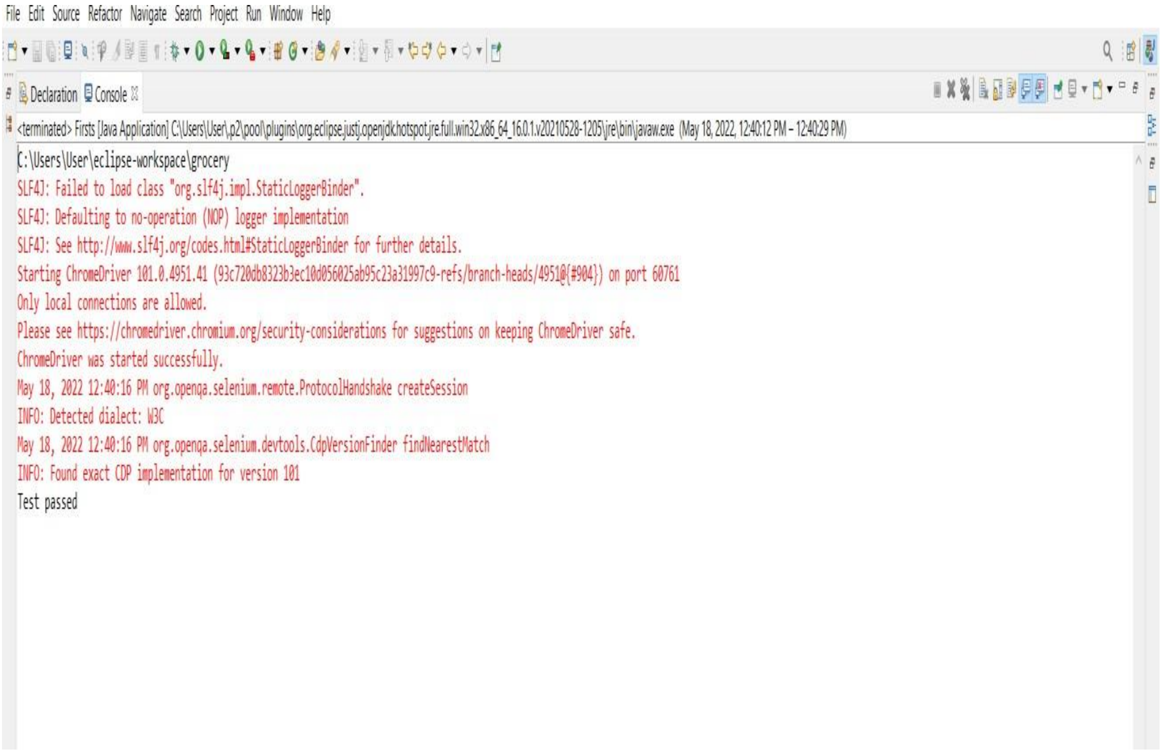
**Post-Condition:** User is validated with database and successfully login into account. The Account session details are logged in database

## Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class UpdatePlayerDetails {
public static WebDriver driver;
public static void main(String[] args) {
driver =  DriverSetup.getWebDriver("http://localhost/bpmssample/login.php");
driver.findElement(By.name("login")).click();
driver.findElement(By.name("email")).sendKeys("sona@gmail.com");
driver.findElement(By.name("password")).sendKeys("sona");
driver.findElement(By.name("submitButton")).click();
driver.get("http://localhost/bpmssample/edit_player.php");
driver.findElement(By.name("address")).sendKeys("Willie House");
driver.findElement(By.name("height")).sendKeys("168");
driver.findElement(By.name("weight")).sendKeys("58");
driver.findElement(By.name("edit")).click();

String actualUrl="http://localhost/bpmssample/player_details.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
```

## Screenshot 5.2.5.3 : Testing 2 Output


File Edit Source Refactor Navigate Search Project Run Window Help
Declaration
Console

```
<terminated> Firsts [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (May 18, 2022, 12:40:12 PM – 12:40:29 PM)
C:\Users\User\eclipse-workspace\grocery
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

# CHAPTER 6

# IMPLEMENTATION

## 6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

## 6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

### 6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### 6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

### 6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## 7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to take ticket booking online by viewing the schedules of tournament. Provides lots of advantages like player and team registration, view profile of players, tournament details, enhanced user interface, payment options, player leave application and add feedback many more.

### 7.2 FUTURE SCOPE

- The proposed system is designed in such a way that it makes the ticket booking online.
- Tournament scheduling can be done online.
- Customers can able to apply for player registration.
- Customers can able to add feedbacks etc.
- Customers can able to view scheduled courses etc.
- Data security can be enhanced.

# CHAPTER 8

# BIBLIOGRAPHY

## REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

## WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://www.slideshare.net/DilipPrajapati4/sport-tournament-managment-system-stms
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

# CHAPTER 9

# APPENDIX

## 9.1 Sample Code

**schedule.php**

```
<?php
include('../config.php');
session_start();
$tr_id = $_GET['rm'];
$s = "SELECT * FROM trm WHERE trm_id=$tr_id";
$q = mysqli_query($con,$s);
while($r=mysqli_fetch_assoc($q)){
$n = $r['trm_name'];
$t = $r['trm_tot_teams'];
$tr = $r['trm_id'];
$_SESSION['trm_id'] = $tr;
} ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<title> Responsive Admin Dashboard</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css"
rel="stylesheet" />
<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>
<link rel="stylesheet" href="../style.css">
<script type="text/javascript" src="../script/option.js"></script>
<link rel="stylesheet" type="../css/option.css" href="">
<!-- Boxicons CDN Link -->
<link href='https://unpkg.com/boxicons@2.0.7/css/boxicons.min.css' rel='stylesheet'>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700
&display=swap');
.container1{
padding-left:14%;
padding-top: 3%;
display:flex;
flex-wrap:wrap;
justify-content:space between;
}
.container1 .box:before{
content:";
width:50%;
height:100%;
position:absolute;
top:0;
left:0;
background:rgbga(255,255,255,.2);
z-index:2;
    }
    .icon img
    .container1 .box:nth-child(odd) .icon{
    box-shadow:0 0 0 0 #e91e63;
    background:#71797E;
    }
    .container1 .box:nth-child(odd):hover .icon{
    box-shadow:0 0 0 400px #71797E;
    }
    .container1 .box:nth-child(even) .icon{
    box-shadow:0 0 0 0 #848884;
    background:#848884;
    }
    .container1 .box:nth-child(even):hover .icon{
    box-shadow:white;
    }
    .container1 .box .content{
    position:relative;
    z-index: 1;
    transition: 0.5s;
    }
    .container1 .box:hover .content{
    color:crimson;
    font-size: 20px;
    }
    .container1 .box .content h3{
    font-size:20px;
    margin:10px 0;
    padding:0;}
    </style>
    </head>
    <body>
         <div class="sidebar">
    <div class="logo-details">
    <i class='bx bxl-c-plus-plus'></i>
    <span class="logo_name">Elite Football</span></div>
    <ul class="nav-links">
    <li>
    <a href="#" class="active"><i class='bx bx-grid-alt' ></i>
    <span class="links_name"onclick="location.href='../dashboard.php';"
    style="cursor: pointer;">Dashboard</span></a></li>
    <li><a href="#"><i class='bx bx-group' ></i>
    <span class="links_name"onclick="location.href='../team.php';"
    style="cursor: pointer;">Teams</span></a></li>
    <li><a href="#"><i class='bx bx-user' ></i>
```

```
<span class="links_name"onclick="location.href='../view_player_a.php';" style="cursor:
pointer;">View Players</span></a></li>
      <li><a href="#"><i class='bx bx-user'></i>
          <span class="links_name"onclick="location.href='../view_coach_a.php';"
style="cursor: pointer;">All Coaches</span></a></li>
      <li><a href="#"><i class='bx bx-user'></i>
          <span class="links_name"onclick="location.href='../add_organizer.php';"
style="cursor: pointer;">Add Organizer</span></a></li>
      <li><a href="#"><i class='bx bx-coin-stack' ></i>
          <span class="links_name"onclick="location.href='booking_details.php';"
 style="cursor: pointer;">Tickets</span></a></li>
      <li><a href="#"><i class='bx bx-coin-stack' ></i>
             <span class="links_name"onclick="location.href='stadium.php';"
style="cursor: pointer;">Stadium</span></a></li>
      <li><a href="#"><i class='bx bx-book-alt' ></i>
          <span class="links_name"onclick="location.href='trmt.php';" style="cursor:
pointer;">Tournament</span></a></li>
      <li><a href="#"><i class='bx bx-group' ></i>
          <span class="links_name"onclick="location.href='../team_coach_map_a.php';"
style="cursor: pointer;">View Coach</span></a></li>
      <li><a href="#"><i class='bx bx-group' ></i>
          <span class="links_name"onclick="location.href='../leave_application.php';"
style="cursor: pointer;">Leave Application</span></a></li>
      <li class="log_out"><a href="#"><i class='bx bx-log-out'></i>
          <span class="links_name" onclick="location.href='../login.php';" style="cursor:
pointer;">Log out</span></a></li></ul></div>
<section class="home-section">
      <nav>
        <div class="sidebar-button"><i class='bx bx-menu sidebarBtn'></i>
            <span class="dashboard">TOURNAMENT LIST</span></div>
        <div class="profile-details"><img src="image/david.jpg" alt="">
                <span class="admin_name">Admin</span></div></nav>
<div class="home-content">
        <div class="overview-boxes">
          <div class="box" onclick="location.href='view_trmt.php';"
style="cursor: pointer;">
            <div class="right-side">
              <div class="box-topic">Back</div></div></div></div>
  <p style="font-family:'Poppins',sans-serif; font-weight: bold;font-size: 25px;
padding-top: 5px;">TOURNAMENT SCHEDULE- <?php echo $n;?></p>
<div class="container1">
      <?php
      function scheduler($teams){
        if (count($teams)%2 != 0){
          array_push($teams,"bye");
        }
```

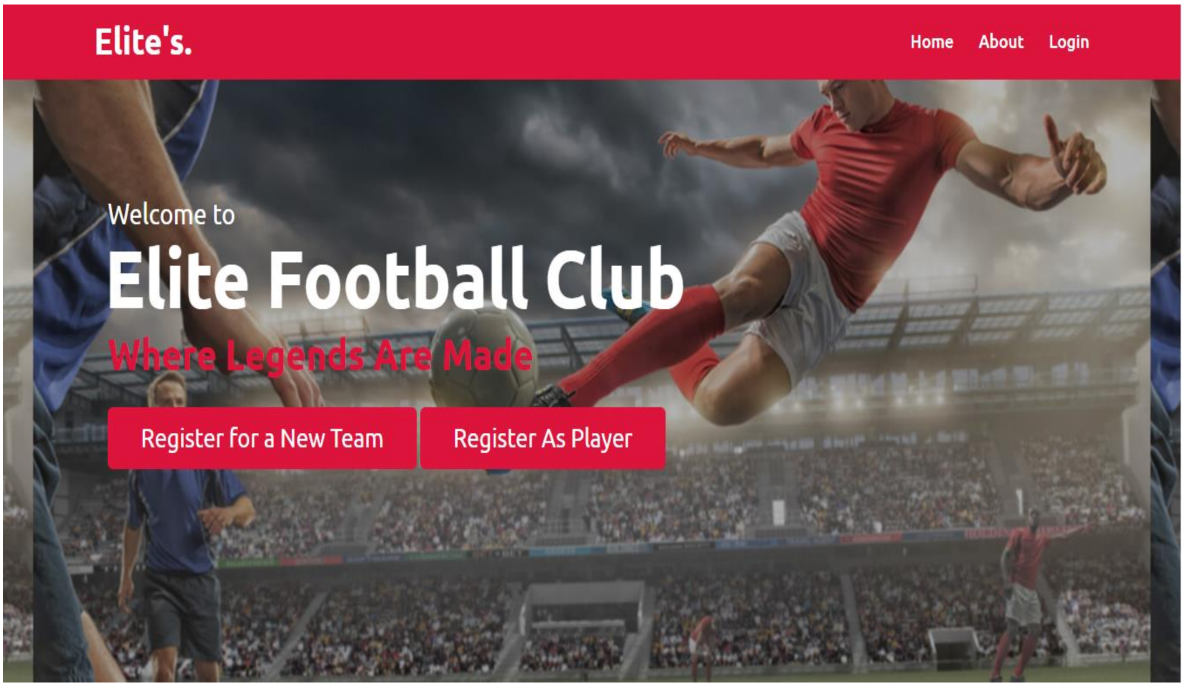
```
$away = array_splice($teams,(count($teams)/2));
$home = $teams;
for ($i=0; $i < count($home)+count($away)-1; $i++){
for ($j=0; $j<count($home); $j++){
$round[$i][$j]["Home"]=$home[$j];
$round[$i][$j]["Away"]=$away[$j];
}
if(count($home)+count($away)-1 > 2){
$tem = array_splice($home,1,1);
$tem1 = array_shift($tem);
array_unshift($away,$tem1);
array_push($home,array_pop($away));
}
} return $round;
}?>
<?php
$members = array();
$sql = "SELECT * FROM `teamlist` WHERE `tl_trmt_id`='$tr_id'";
$query = mysqli_query($con,$sql);
while($res = mysqli_fetch_assoc($query)){
$te_id = $res['tl_team_id'];
$sql2 = "SELECT `teamr_name` FROM `team_reg` WHERE `teamr_id`='$te_id'";
$result = mysqli_query($con,$sql2);
if(mysqli_num_rows($result)>0)
{
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
array_push($members, $row["teamr_name"]);}}

else { echo "0 results";}}        ?>
<?php $schedule = scheduler($members); ?>
<?php
$member = array();
foreach($schedule AS $round => $games){
$ro = ($round+1);
echo "<a href='trmt_time.php?ttm=$ro'><div class='box'>Round: ".($round+1).";
echo "<div class='content'><br>";
foreach($games AS $play){
echo $play["Home"]." - ".$play["Away"]."<BR>";
$h = $play["Home"];
$a = $play["Away"];
if($t % 2 == 0){
$count = $t/2;
}else{
$count = ($t+1)/2;
}
$sqlh = "SELECT `teamr_id` FROM `team_reg` WHERE `teamr_name`='$h'";
$resulth = mysqli_query($con,$sqlh);
while($resh = mysqli_fetch_assoc($resulth)){
$th = $resh['teamr_id']; }
$sqla = "SELECT `teamr_id` FROM `team_reg` WHERE `teamr_name`='$a'";
$resulta = mysqli_query($con,$sqla);
while($resa = mysqli_fetch_assoc($resulta)){
$ta = $resa['teamr_id']; }
$sqltr = "SELECT * FROM `t1_vs_t2` WHERE `tl_trmt_id` = '$tr_id' AND
`tv_round` = '$ro'";
$querytr = mysqli_query($con,$sqltr);
if(mysqli_num_rows($querytr)>=$count)
{}
else{
if($h == "bye"){
$th = 0; }
else{$h = $play["Home"];}
if($a == "bye"){
$ta = 0; }
else{$a = $play["Away"];}
$sqlah = "INSERT INTO `t1_vs_t2`(`tl_trmt_id`,`tv_round`, `tv_home_id`,
`tv_away_id`,  `tv_date`, `tv_start_time`, `tv_end_time`, `tv_venue`) VALUES
('$tr_id', '$ro','$th','$ta',0,0,0,0)";
if(mysqli_query($con,$sqlah)){
//echo "Registered successfully";}
else{
echo mysqli_errno($con); }}}
echo "<BR></div></div></a>";}  ?>
</div></div></section></center>
<body>
</html>
```

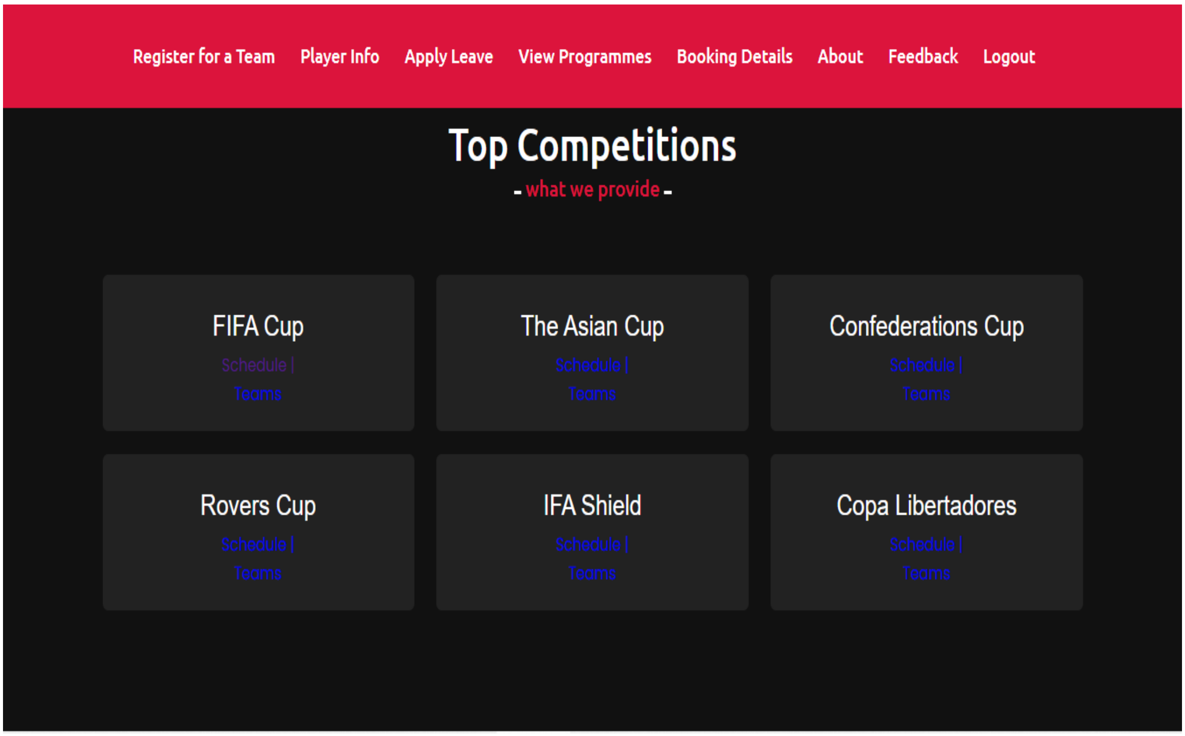
## 9.2 Screen Shots

### CUSTOMER PAGES

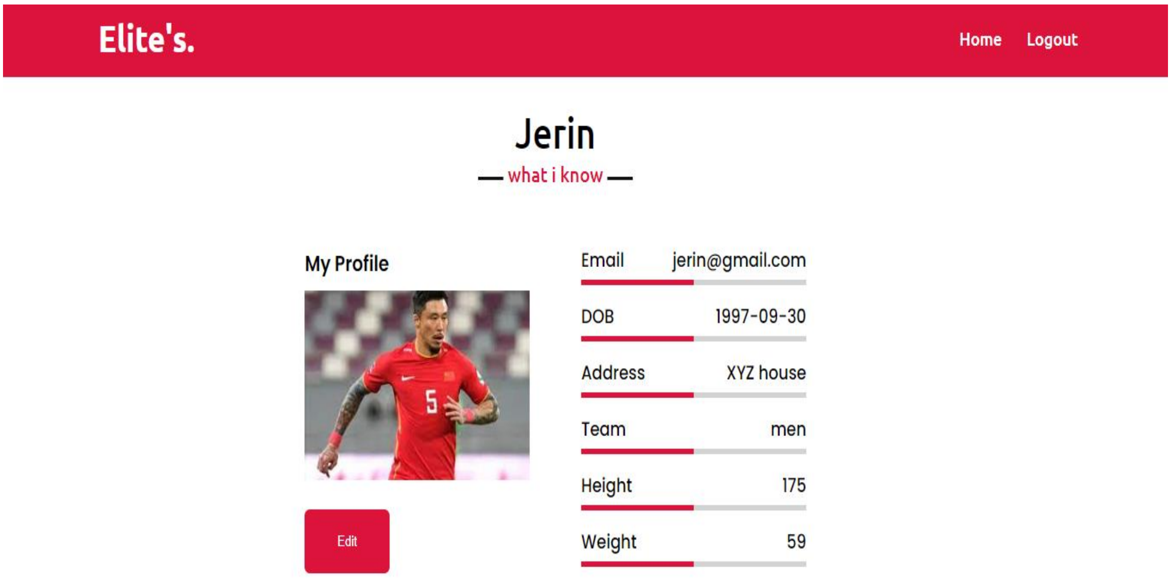
**Screenshot 9.2.1 : Customer Home Page**



**Screenshot 9.2.2 : Customer Tournament View page**



### Screenshot 9.2.3 : Player Details page



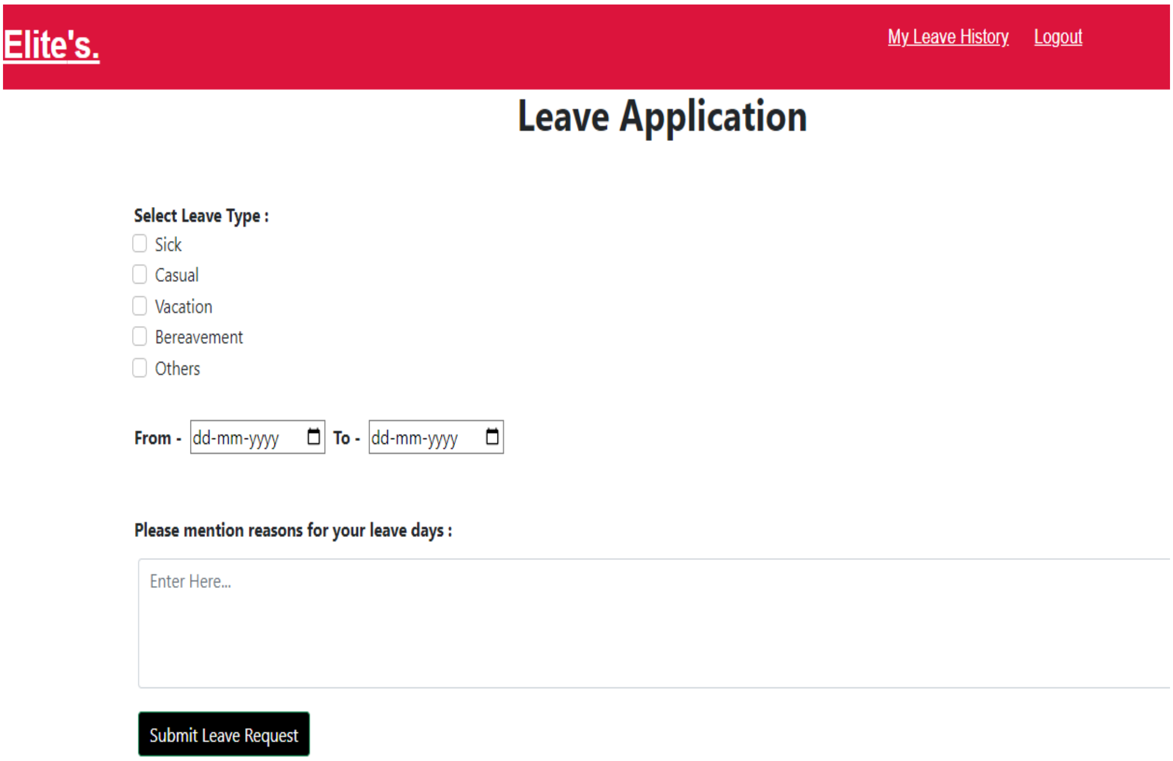
Jerin

— what i know —

My Profile

| | |
|---|---|
| Email | jerin@gmail.com |
| DOB | 1997-09-30 |
| Address | XYZ house |
| Team | men |
| Height | 175 |
| Weight | 59 |

### Screenshot 9.2.4 : Player details editing page



Leave Application

Select Leave Type :

□ Sick

□ Casual

□ Vacation

□ Bereavement

□ Others

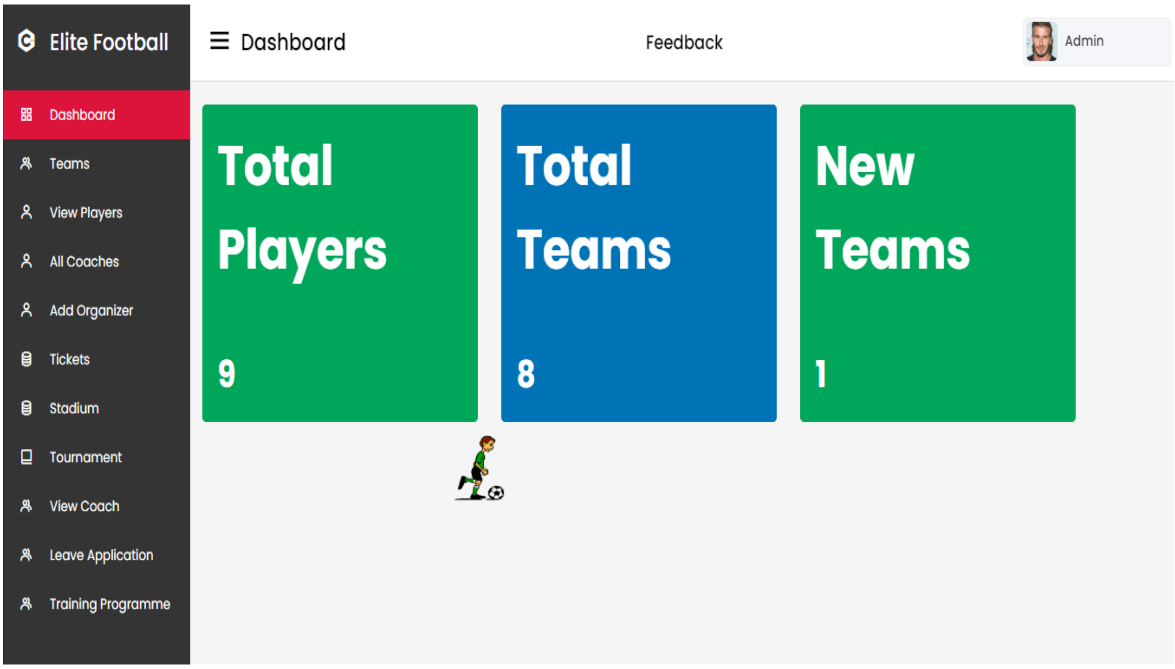
From - dd-mm-yyyy To - dd-mm-yyyy

Please mention reasons for your leave days :

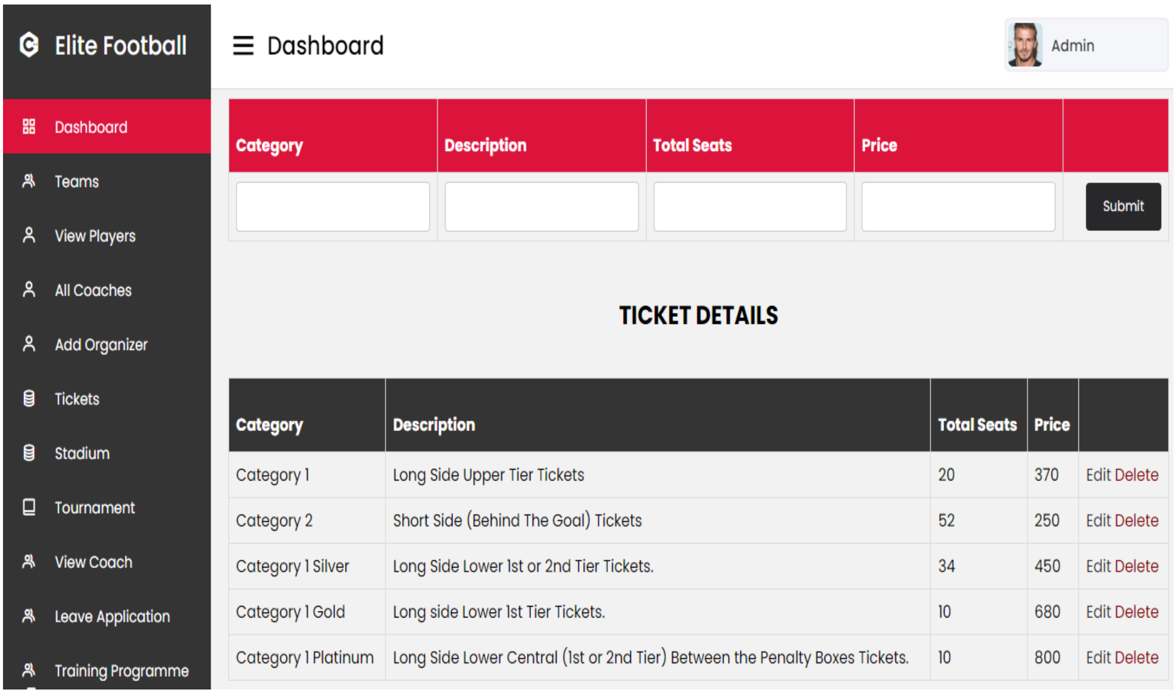
Enter Here...

## ADMIN PAGES

### Screenshot 9.2.5 : Admin Dashboard Page



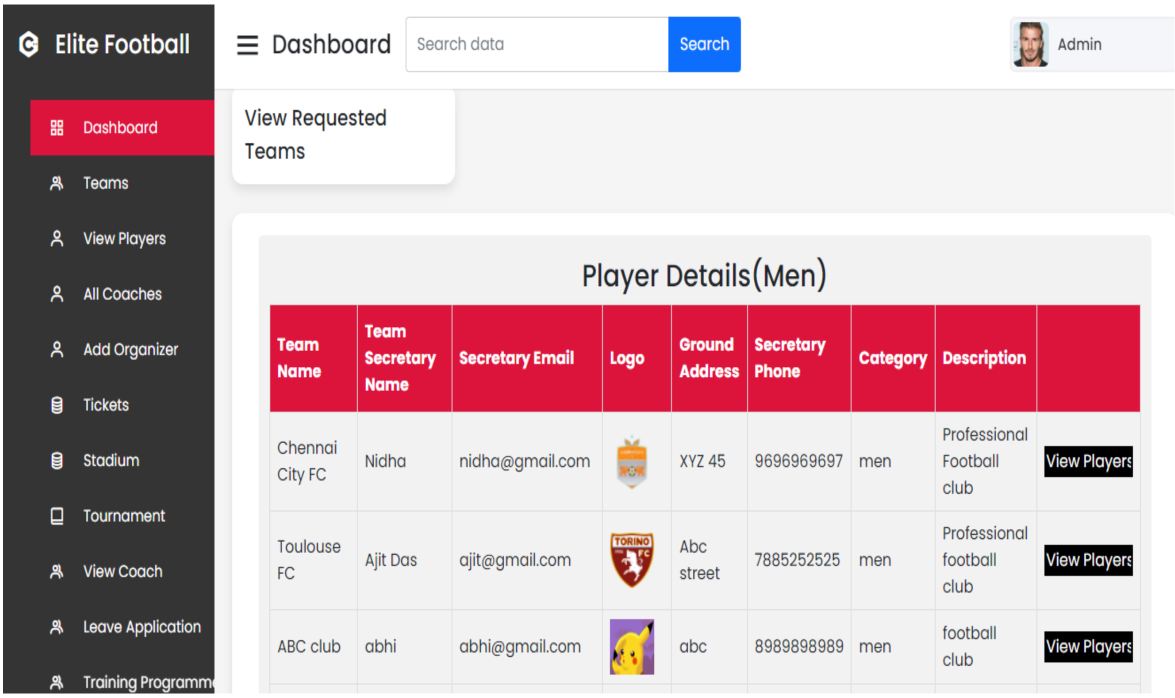
### Screenshot 9.2.6 : Ticket Details Page



TICKET DETAILS

| Category | Description | Total Seats | Price | |
|---|---|---|---|---|
| Category 1 | Long Side Upper Tier Tickets | 20 | 370 | Edit Delete |
| Category 2 | Short Side (Behind The Goal) Tickets | 52 | 250 | Edit Delete |
| Category 1 Silver | Long Side Lower 1st or 2nd Tier Tickets. | 34 | 450 | Edit Delete |
| Category 1 Gold | Long side Lower 1st Tier Tickets. | 10 | 680 | Edit Delete |
| Category 1 Platinum | Long Side Lower Central (1st or 2nd Tier) Between the Penalty Boxes Tickets. | 10 | 800 | Edit Delete |

## Screenshot 9.2.7 : Team Details Page

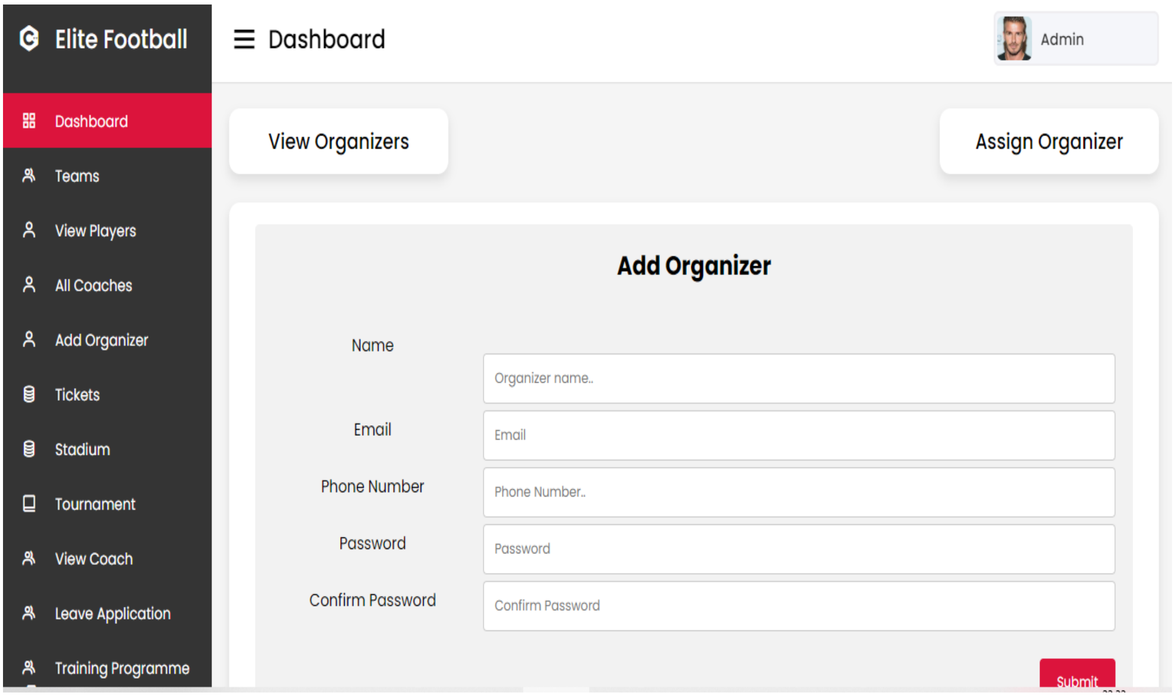


Player Details(Men)

| Team Name | Team Secretary Name | Secretary Email | Logo | Ground Address | Secretary Phone | Category | Description | |
|---|---|---|---|---|---|---|---|---|
| Chennai City FC | Nidha | nidha@gmail.com | | XYZ 45 | 9696969697 | men | Professional Football club | View Players |
| Toulouse FC | Ajit Das | ajit@gmail.com | | Abc street | 7885252525 | men | Professional football club | View Players |
| ABC club | abhi | abhi@gmail.com | | abc | 8989898989 | men | football club | View Players |

Screenshot

## Screenshot 9.2.8 : Add Organizer Page



Add Organizer

Name: Organizer name..

Email: Email

Phone Number: Phone Number..

Password: Password

Confirm Password: Confirm Password