# CH 3 Design

Design or system design helps in identifying hardware and system requirements and also aids in defining overall system architecture. Structural design, Behavior design, Database design, Architecture design are some example that arise in this phase. Design phase has become important as design phase has significant impact on implementation stages. A good design can help the system significantly.

# Structural Design

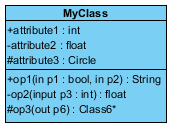
Structural design is the method to find the protected, durable and economical requirement of the structure or system which include materials, technology, and geometry, the extent of the members necessary to complete the structure or system on certain life period of the structure. (Monalisa patel, 2019). Some example are Class diagram, DFD to Flowcharts, etc.

## Class Diagram

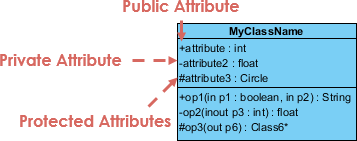
Class Diagram is a static diagram that explains the attributes and operation of a class and also the constraints imposed on the system. A class Diagram displays a group of classes, interfaces, association collaboration and constraints

The importance of class diagram are as follows:

* Analysis and design of the static view of a system.
* Explain responsibility of a system

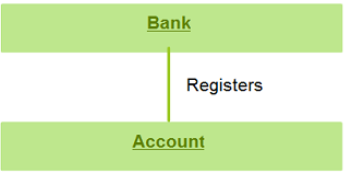
**UML class Notation**

* **Class Name:** Class name are shown in the first partition that
* **Class Attributes:** Class Attributes are shown in second partition that shows the characteristic of a class that is of interest for the user of the system.
* **Class Operation:** Class operation are shown in the third partition which explains the facilities that the class provides
* **Class Visibility:** Class visibility are the symbol before any operation and attributes name in a class denote the visibility if the operation and attributes. The symbol are as follows + which indicates public, - which indicate privateand # which indicates protected attributes and operation.

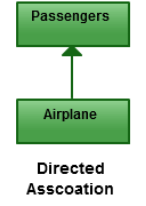
****

**Class Relationship:** one class can be involved with one or more than one relationship with other classes. There are many relationship types they are as follows:

* **Simple Association:** When two classes are linked to each other in any ways with a solid line then simple association relation is established.

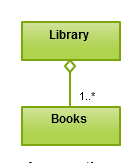
****

* **Directed Association:** Directed association is displayed between class with a solid line with an arrow that indicates the direction of navigation

****

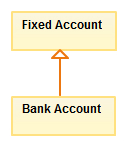
* **Aggregation**

Aggregation are mentions to the formation of a certain class as a result of one class being aggregated or built as a collection where the contained classes are not strongly depended on the lifecycle of the container.



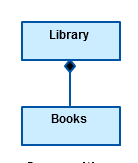
* **Generalization:**

Generalization are also known as inheritance where the relationship wherein one associated class is a child of another by the same functionalities of the parent class.



* **Composition**:

Composition relationship is very similar where there is a strong lifecycle dependency between the contained class and the container



Here is the final class diagram for Football league Management System.

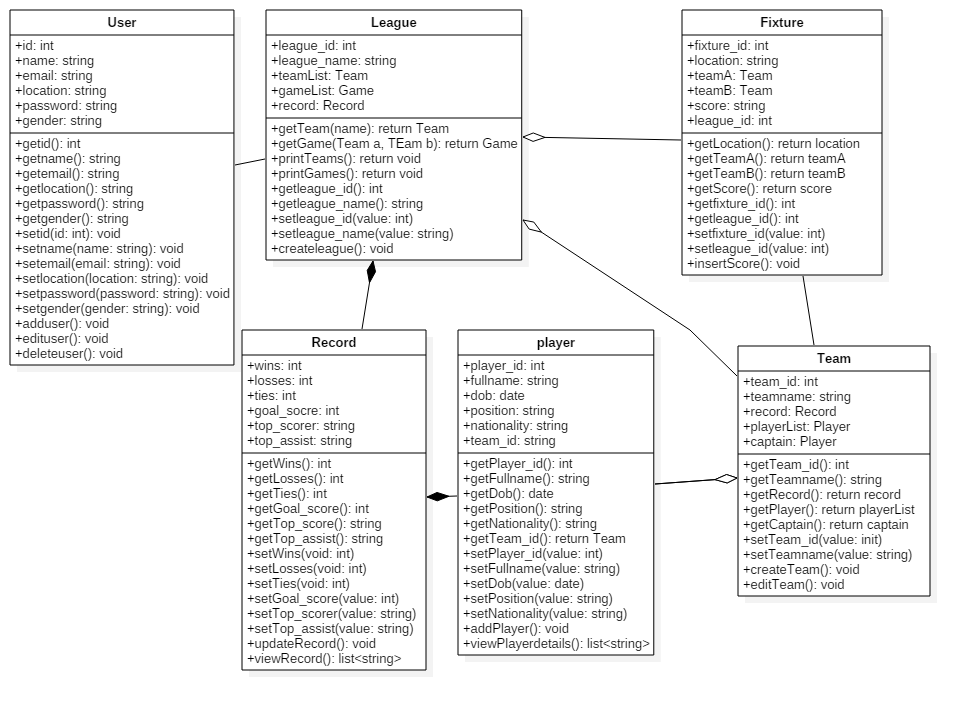
****

Figure 1 Final Class Diagram

The following class diagram shows the relationship between different classes. As shown in figure user can register and can view league, team, player, record and fixture. The league aggregation relationship with fixture and team meaning deleting of league doesn’t delete other and has composition relation with record meaning deleting of league will also delete the record of the league same as between player and record. Player and team has aggregation relation between them as player doesn’t get deleted if the team get deleted. This class diagram shows how one class affect the other.

## Flowchart

Flowchart is a graphic illustration of the sequence of step and decision needed to perform a process. A flowchart shows the workflow or the process or also demonstration algorithm, a step-by-step approach to solving a task.

Importance of flowchart are

* Flowchart provide better way of communicating the logic of the system
* Problem can be analyzed in more effective way which can save cost and time
* It help as guide or blueprint during the program development phase etc.

**Notation used**

* **Start/End symbol:**

An oval represent ta start and end point.



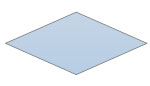
* **Action or process symbol:**

A rectangle represent a process or single step within a larger process



* **Decision Symbol:**

A diamond shows decision symbol explain different decision emerge from different points



* **Input/output Symbol:**

A parallelogram shows input or output represent material or information entering or leaving the system.



* **Arrow:**

Arrow is a line of connector that shows relationship between representative shapes.

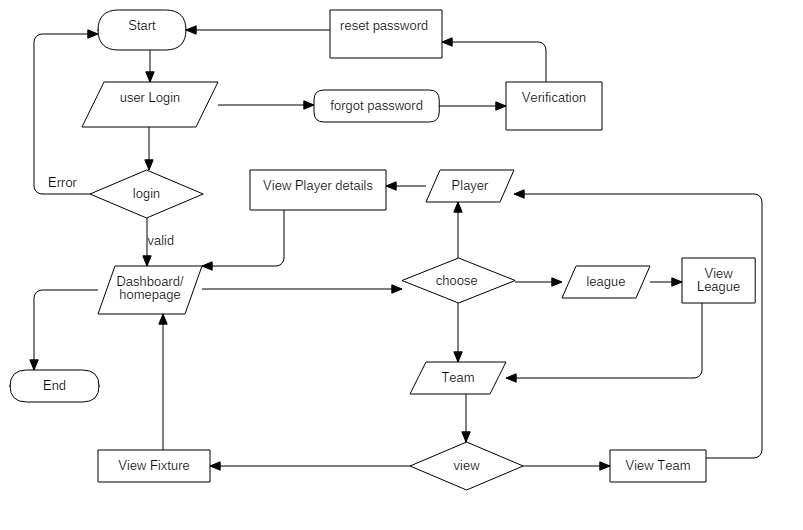


Figure 2 flowchart for system

The flowchart shows how the user interact the system from login. The flowchart shows how step by step of how user can use the system if they like to choose between to view, league or team or player information and to view fixture. The homepage contains news and

# Behavior Design

## Activity Diagram

Activity diagram shows the flow of activity from one to another. It can be explains as an operation of the system where it drawn as the control flow from one operation to another. I

Purpose of activity diagram are as follows:

* To draw the activity flow of the system
* To display the message flow from one activity to another
* To explains the sequence from activity to another

**Notation of activity Diagram**

* **Initial state or Start Point:**

It is the first column which represent the start point of the diagram which is shown by a small filled circle followed by the arrow.



* **Activity or Action State:**

The Activity or action state like in name signifies the non-interruptible action or activity of the object. It is shown or drawn by using rectangle with rounded corners

* **Action Flow:**

Action flows are represented by arrowed lines that shows the directional flow, or connect flow of the activity.



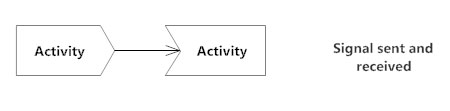
* **The decision symbol:**

The decision symbol is a diamond shape which signify the branching or merging of various flows with the symbol acting as frame or container. It is used when a activity requires a decision before moving to the next step



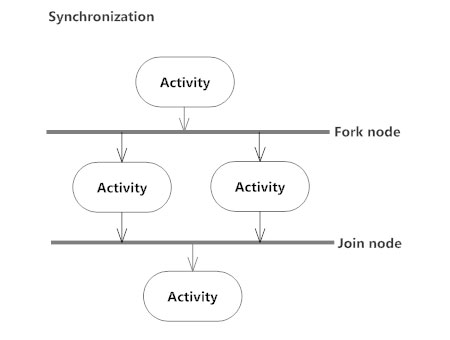
* **The sent and received signal:**

The send signal shows that the signal is being sent to a receiving activity as the received signal shows the acceptance of an event send from the sent signals



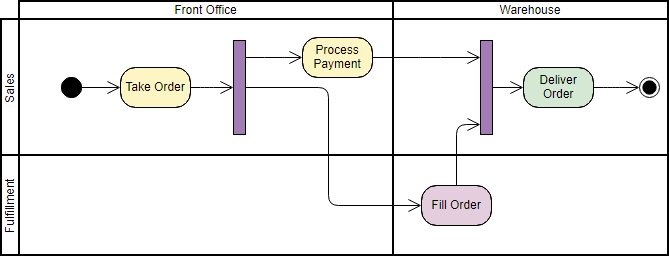
* **Synchronization** **Bar:**

Join node is a thick vertical or horizontal lines that combines multiple concurrent flows back into single outgoing flow and Fork node is that split a single incoming flow into multiple concurrent flows. Both fork and join node which are used together are often referred as synchronization.



* **Swimlane:**

A Swimlane show clarity and accountability by placing a process steps within the horizontal or vertical lanes that provides information about the connection, communication and handoffs between these lanes.



* **Final State or end point**

End Point represent the completion of a process which is represent by an arrow pointing to a filled circle nested inside another circle.

****

Here are some of the activity diagram

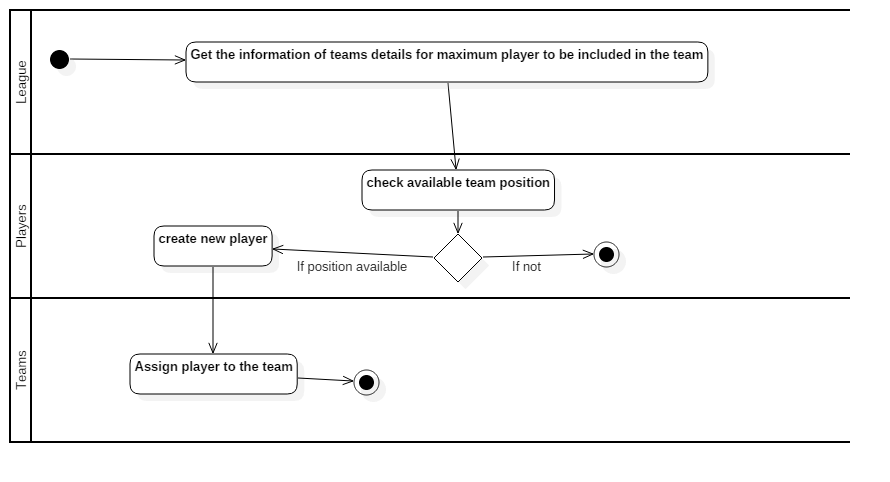


Figure 3 Activity diagram (ADD new Player)

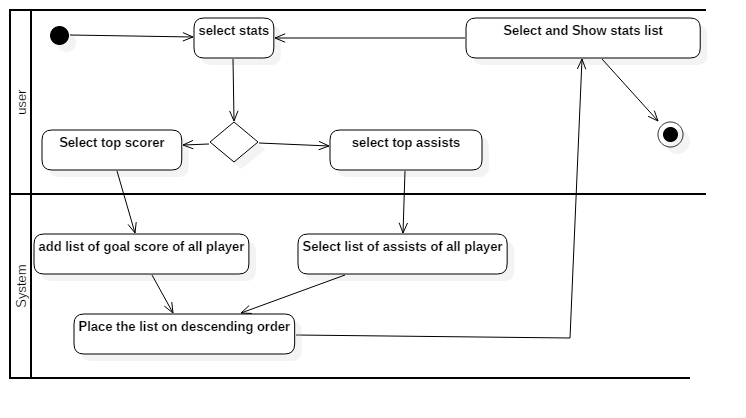


Figure 4 View Player Record

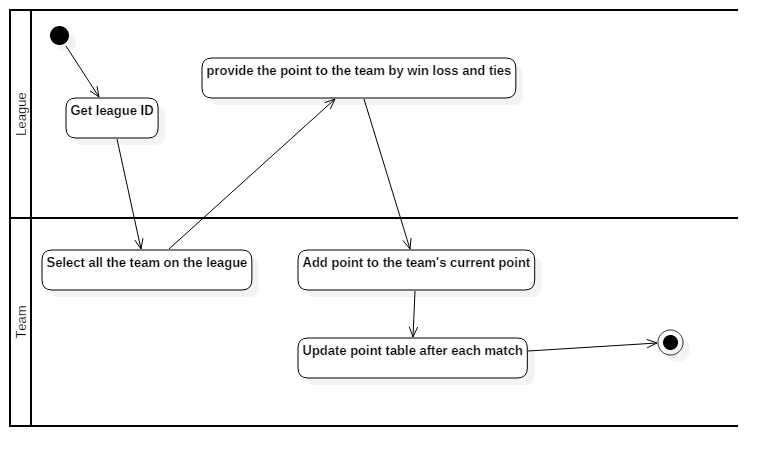


Figure 5 League Table

The above activity diagram shows the activity of how the system works. It provide information about some key features and shows step by step activity to complete the process. The first diagram shows small activity of how to add a new player like from getting information about the team to checking team position available then to create new player or choosing different team if there is no position available to creating new player and then assigning new player to the team. That same goes for the second figure which shows activity process to view player stats.by going on stats and then choosing between player top goal assist or top scorer then the system get the list of player stats and shows the player stats. And third one shows how the league table point keep on updating.

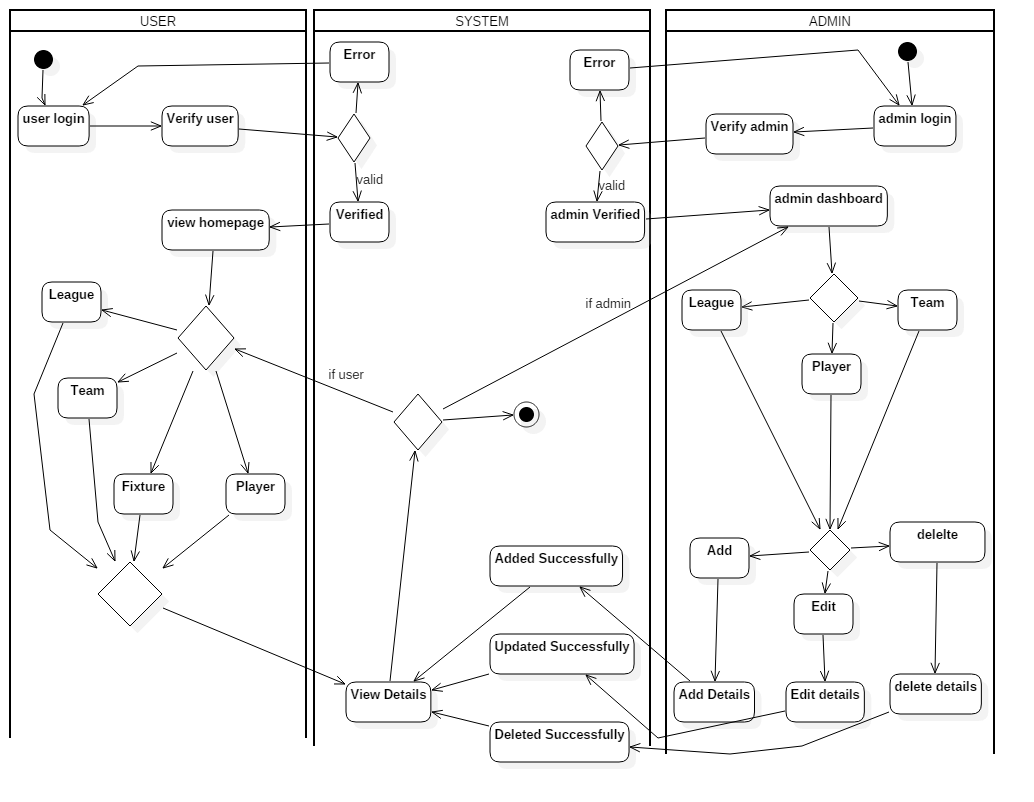


Figure 6 full admin or user using system

The figure shows the activity of login between user and admin and how user and admin can used the system. It shows activity of admin and how admin can add, edit or delete player and saved to the system and how user can view the details saved by the admin in the system.it shows if the user or admin after completing certain activity and want to end the activity or restart the process. To view or add different player, team, league etc.

## Sequence Diagram

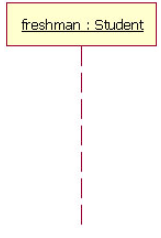
A sequence diagram are interaction illustrations that displays information on how the operation are carried out. Sequence diagram shows the interactions between objects in the context of a collaboration. It shows what message are sent and when. It is organized according to time.

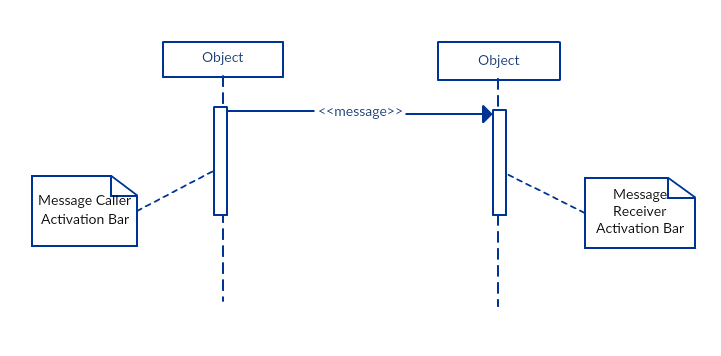
The purpose of sequence diagram are as follows:

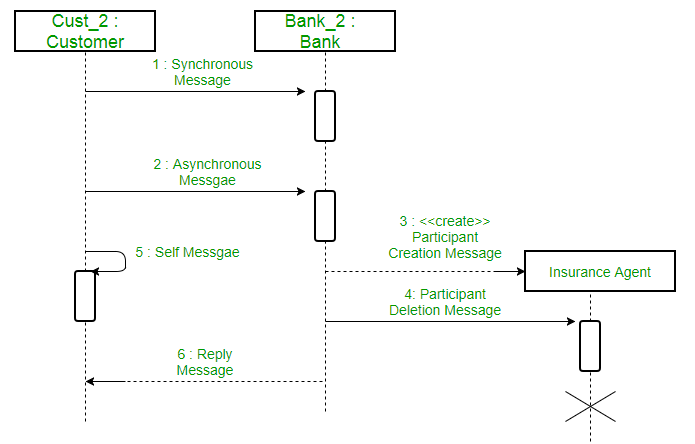
* To shows the interaction between object in the chronological order that those interaction occur.
* It shows how the system works by showing how different system object interact.

Notation of sequence diagram are as follows:

* **Lifeline**: Lifeline are illustrations as a box with a dashed line arranged horizontally across the top of the diagram.

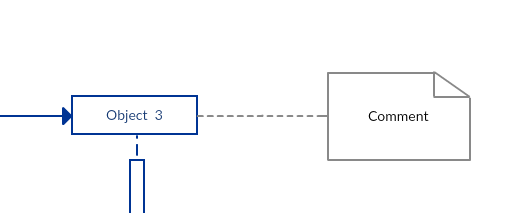


* **Activation** **Bar**: Activation bar are illustration as a box placed on the lifeline. It shows that the object is active during the interaction process between two object.
* **A message arrow:** A message arrow is illustration as a line sending a message from an object or lifeline to the receiving object with the solid arrowhead.

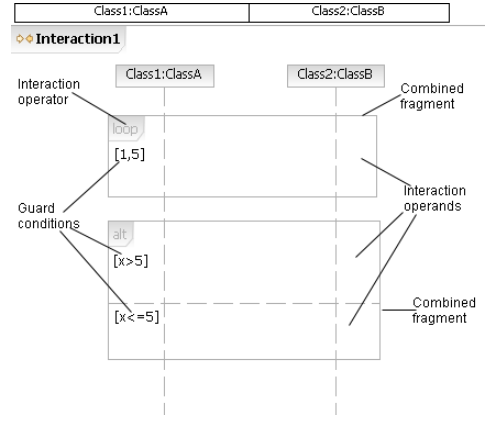


As shown in above figure there are different types of message which are explain as bellows

* **A synchronous message:** A synchronous message is a solid arrow heads message where the sender has to waits till the receiver process and return the message to continue to another message to other object within the system
* **A asynchronous message:** A asynchronous message is also a solid arrows head message where the sender doesn’t has to waits till the receiver process and return the message before sending other message to other object within the system
* **Return or Reply message:** A return messages is used to specify that the message receiver is done processing the message and is returning control over to the message caller.
* **Self: Message:** A self-message is used to specify when an object send a message to itself which is shown with a message arrow that starts and ends at the same lifeline..
* **Comment:** Comment is a rectangle with a folder-over cornet which is linked to the related object with a dashed line

****

* **Sequence fragments:** Sequence fragment also known as combined fragments which are the logical grouping represented by rectangle which contain the condition structure that affect the flow if message



Here are some of the Sequence diagram for football league Management System

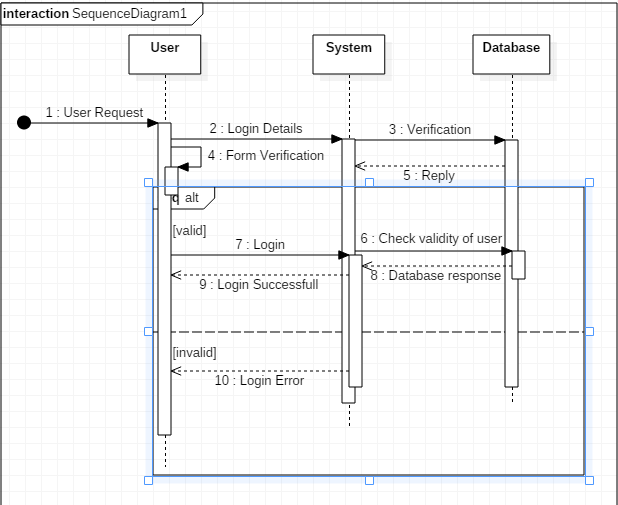


Figure 7 Sequence Diagram Login

The Sequence diagram shows how user login operation carried out. Above diagram shows step by step process

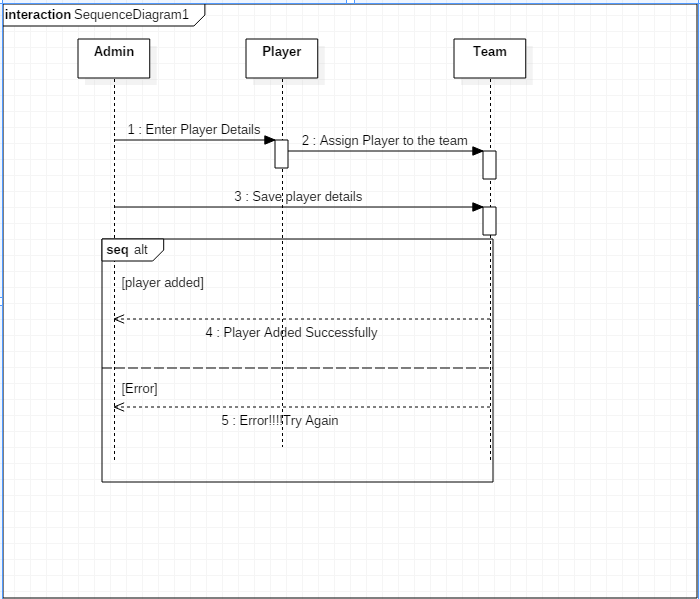
****

Figure 8 Sequence Diagram New Player Added

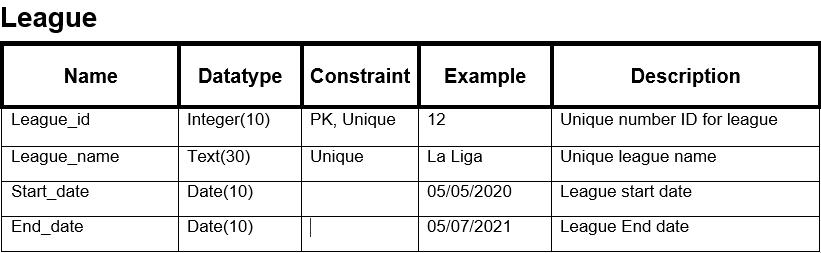
# Database Design

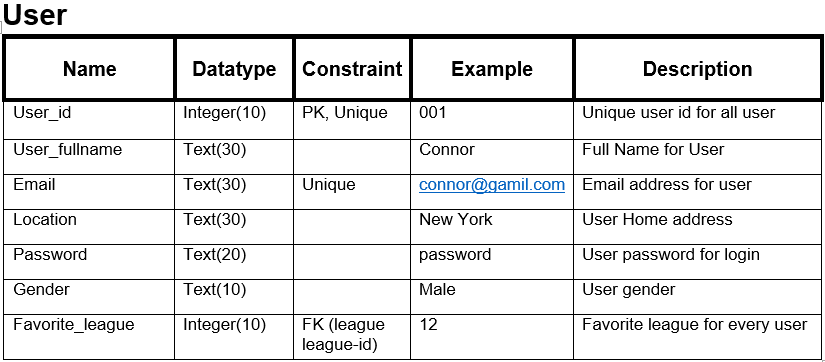
## Data dictionary

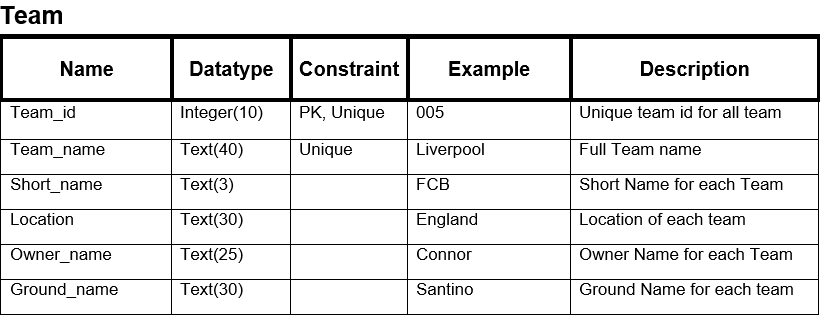
Data dictionary is explains as the dictionary about the data that are store in the database. A data dictionary holds all the info about the data object

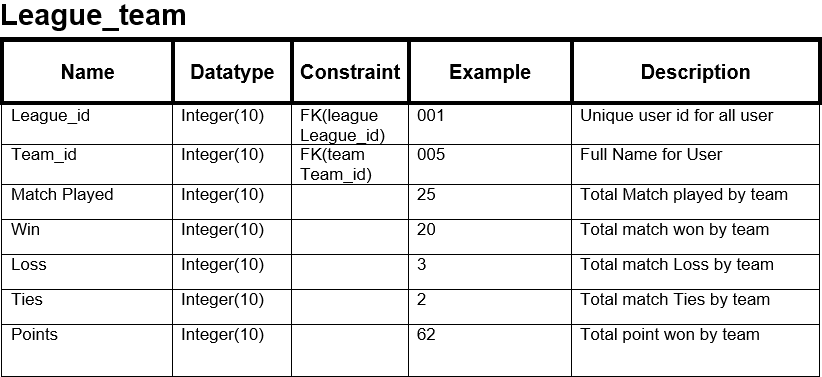
A data dictionary is importance because

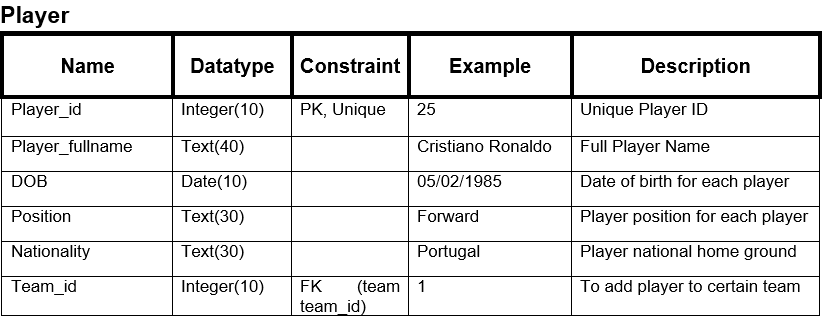
* It make easier to identify and understand the factors about the object
* It provides following information:
* Names of all tables in database and their owners.
* Names of all indexes and the columns to which the tables in those indexes related.
* Constraints defined on tables, including key relationships to other tables and constraints.

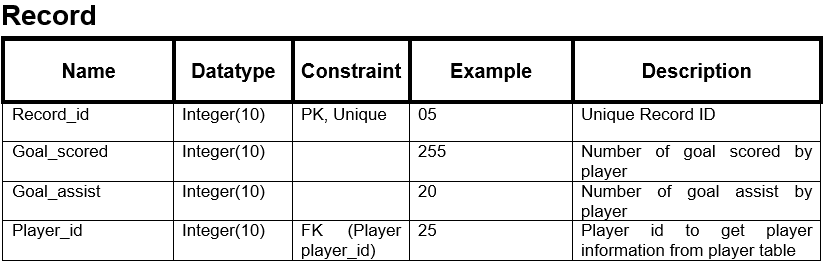






\







The following data dictionary has provided information about all the data related to each table their datatype, relation and constrain. Every data on table above is not null able and need. The table also shows relationship between two tables and how one affect another. With given example it easier to get information.

## ER diagram

ER diagram stands for Entity relationship diagram that is use in database design. ER diagram shows the connection between user/admin, object, place, concept or event within the system as it is the graphical representation of an information system.

The importance of ER diagram are as follows;

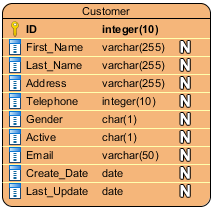
* Database design can be well design and easy to understand
* Shows the interlinked between tables including their cardinality/relationship

**Notation used on ER diagram are as follows:**

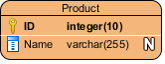
* **Entity**: Entity are things or concept within a system. Entity are also knowns as nouns. The image shown as is an example of an entity

. 

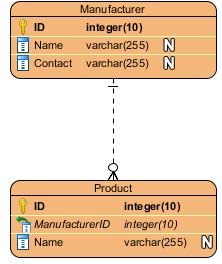
* **Entity Attributes:** Entity attributes are the column which shows the property or specific of the entity that holds it

. 

* **Primary key:** Primary key are special type of entity attributes that unique describes a data in a database table.

****

* **Foreign key:** Foreign key is the reference of the primary key in a table.

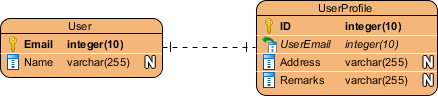
****

* **Relationship (Cardinality)**

Cardinality refers to the possible number of occurrence in one entity which are associated to the number of occurrences in another.

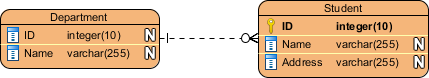
* **One –to-one cardinality:**

In one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable.



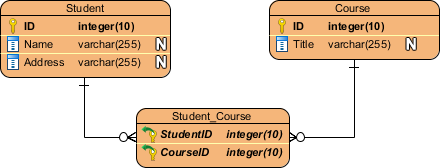
* **One-to-Many cardinality:**

A one-to-many relationship states to the relationship between two entities in which an instances of one entity example x may be linked to many instance of other entity example y but an instance of y is only linked to one instance of x.



* **Many-to-Many cardinality:**

A many-to-many relationship states to the relationship between two entities in which both entities may be linked to many instances of each other which is split into a pair of one-to-many relationship in physical ERD.



Here is the ER Diagram of Football league management system.

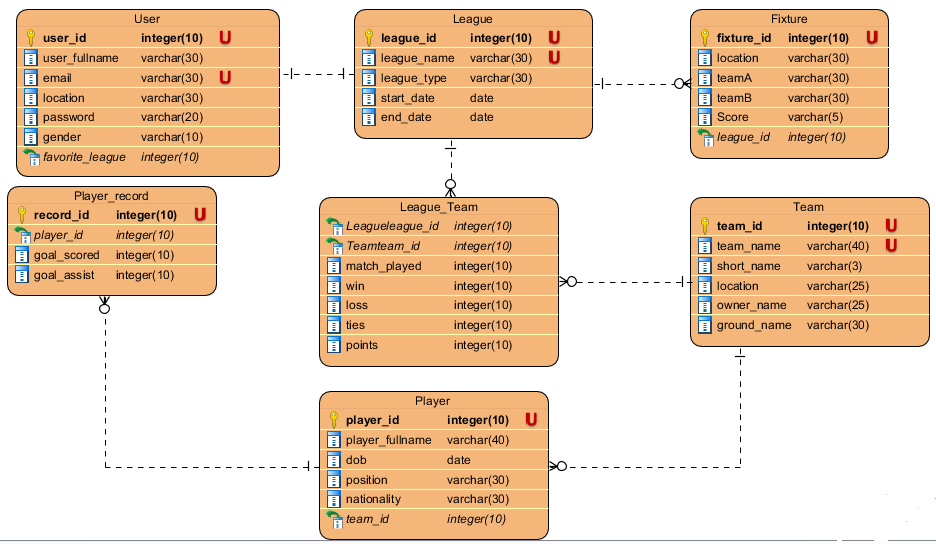


Figure 9 ER\_Diagram

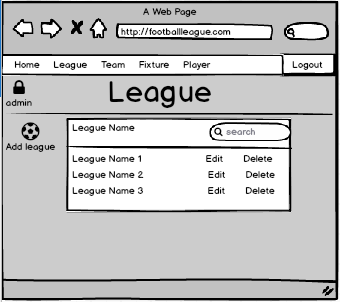
# Architecture Design

## UI: Prototyping

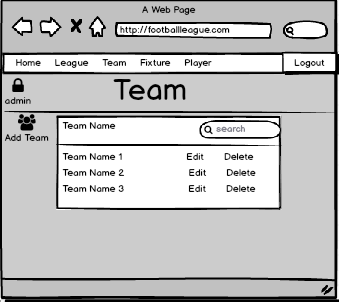
A prototype is an early sample, model, concept or process. Digital prototype is a digital simulation or demo which provide conceptual design that help to virtually explore a complete product before it’s built. It help to provide information about what the system might look liked and how the system works and to check if the product works.

Some UI Digital prototype are as follows

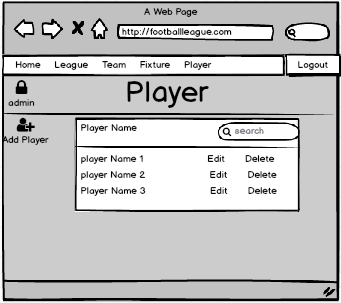
 Admin login helps to login admin so they can add and update league, teams, player fixture, News, etc.



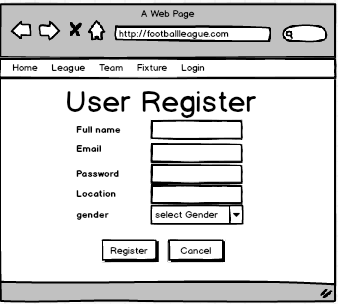
Admin can add new league delete or update all the league.



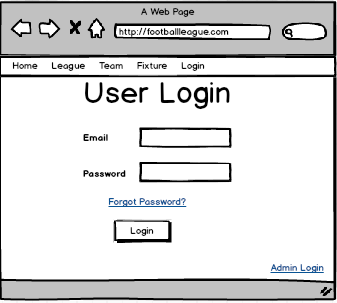
Admin can add team to the league, edit or remove the team



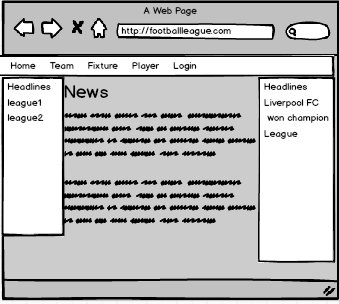
Admin can add new player to all the teams. They can add player stats update them.



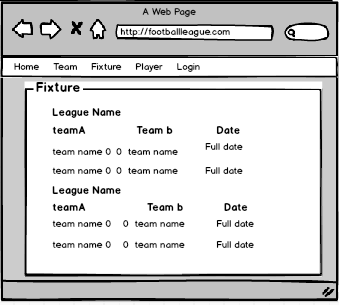
User can register their profile so they can get more content



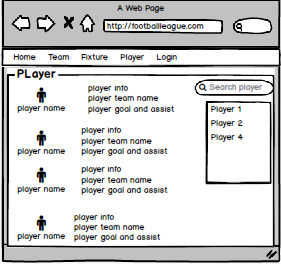
User can login to access their profile and get more content to the page



Home page of page contains news and headlines related to major event happening on the football world



Fixture page has all the league teams’ match and date so user can get information about them.



Player information can provide player details to the user and their goals and assist.

# Conclusion

Football league management system with the completing of design can help in implementation phase. All above design has given information about how the system works, flows. It has provided database design which has provided all the table required for this project. It has also provide information on all related activity on the project. The process and flow of what user and system action for certain task. With digital prototype it provide what the system might look like and how the system works from one page to another.