# Acknowledgement

I would like to thanks my subject teacher for this opportunity to let me demonstrate my idea and display it to the other. I can show my ideas and have a good overview as well as it helped me to extend my knowledge about the language and I encountered new problems as well as solution that increased my skill.

Contents

[Acknowledgement 1](#_Toc15287354)

[Chapter 1: Overview 4](#_Toc15287355)

[1. Introduction 4](#_Toc15287356)

[2. Background 4](#_Toc15287357)

[3. Description 4](#_Toc15287358)

[Scope of the project 5](#_Toc15287359)

[1. Scope 5](#_Toc15287360)

[2. Aims 5](#_Toc15287361)

[3. Limitation 5](#_Toc15287362)

[4. Objectives 5](#_Toc15287363)

[5. Overview of the project 5](#_Toc15287364)

[Development Methodology 6](#_Toc15287365)

[1. Waterfall model 6](#_Toc15287366)

[2. Design pattern 7](#_Toc15287367)

[3. System architecture 8](#_Toc15287368)

[Project planning 8](#_Toc15287369)

[1. Work Breakdown Structure 8](#_Toc15287370)

[2. Milestone 9](#_Toc15287371)

[Figure 1 : time estimation 9](#_Toc15287372)

[3. Scheduling 10](#_Toc15287373)

[3.1 Time estimation 11](#_Toc15287374)

[3.2 Gantt Chart 12](#_Toc15287375)

[Risk Management 13](#_Toc15287376)

[Configuration management 15](#_Toc15287377)

[Chapter 2: Analysis 16](#_Toc15287378)

[Feasibility study 18](#_Toc15287379)

[Requirement analysis 19](#_Toc15287380)

[Functional 19](#_Toc15287381)

[Non-functional 20](#_Toc15287382)

[MOSCOW Prioritization 21](#_Toc15287383)

[Use Case 23](#_Toc15287384)

[NLA 25](#_Toc15287385)

[Assuming Our Scenario 25](#_Toc15287386)

[Unfiltered Selection 25](#_Toc15287387)

[Filtered Selection 25](#_Toc15287388)

[Chapter 3. Design 26](#_Toc15287389)

[3.1 Introduction 26](#_Toc15287390)

[3.2 Structural Design 27](#_Toc15287391)

[a) Class diagram 27](#_Toc15287392)

[b) Data flow diagram (DFD) 28](#_Toc15287393)

[3.3 Behavioral Diagram 30](#_Toc15287394)

[a. Sequence diagram 30](#_Toc15287395)

[b. Activity diagram 32](#_Toc15287396)

[3.4 Database management system 34](#_Toc15287397)

[a. Data Dictionary: 34](#_Toc15287398)

[b. ER (Entity relationship): 35](#_Toc15287399)

[3.5 Architecture 37](#_Toc15287400)

[Chapter 4: Implementation 38](#_Toc15287401)

[Chapter 6: Issues 69](#_Toc15287402)

[Chapter 7:Conclusion 70](#_Toc15287403)

# Chapter 1: Overview

## Introduction

League manager is a project I have choose to work on. It can be a very reliable means of organizing a competitive football league. With this, there will be less error in the league and everything will be free fair and transparent as well. It mainly focuses on league system rather than the knockout competitive tournaments. It can help you with time management and the use of paper and pens as well.

## Background

Football is the most loved and played game in the world. The love and passion for it has never decreased and neither will it. Every country has a league of its own. To organize a league it is very messy work. Data can be lost misused and hard to handle. So League organizer can help people a lot in organizing a league.

## Description

The reason why this project is done is to eradicate the problems we have while organizing a league using pen and a paper. League can take a lot of time so managing them is a really hard work but no more with this project. The old days are gone.

The expected features of this project are:

1. Organizer can add description about team.
2. They can add teams and players.
3. They can properly add details about players.
4. The table will be updated automatically.
5. Players with cards will be stored as well as goals.
6. Players will be banned as per the rule.
7. Fixtures can be shown easily.
8. Champions will be declared automatically.

## Scope of the project

## Scope

Scope is that part of project, which involves detailed information about the project such as goals, task, features etc. Documentation explains boundaries of the project as well as procedures and establishes responsibility for team members.

## Aims

* 1. Easy to manage leagues.
  2. Easy to use.
  3. Less error

## Limitation

1. It is not errorless.
2. It hasn’t reached its full potential
3. Many further features could be added.
4. The user interface may not be applicable for everyone.

## Objectives

To meet the aim of the project objectives are listed below:

1. To organize a league properly.
2. To automate the results.
3. Easy to use application.
4. View results and fixtures.
5. To view players detain in the league.
6. To have an automated table.

## Overview of the project

The limitation of this project is that it is only a league-based project rather than other type of project like knockout competition and it focuses on football only not other type of sports. It is very easy to use and very reliable too. It is quick and automated which removes the traditional way of organizing a league.

## Development Methodology

## Waterfall model

For our project to be a successful, we need to follow development methodology because it develops a project systematically in a serial way rather than in a messy way.it can help our project to be more productive and helps in time management as well. Among all the development methodology like waterfall and agile I have chosen waterfall Methodology for this project.

Waterfall method is that type of model in which we move on to another process after the completion of the current process. Because our requirements are known already and we do not have any further change in the requirement cause the project is short as well which means we do not have any ambiguous requirement . This method has very little amount of disadvantage when used in cases like ours.

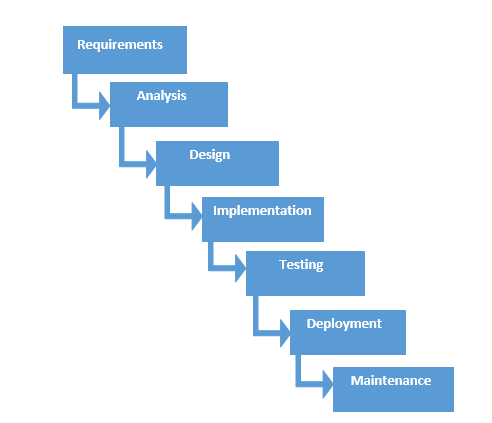


Figure 1: Waterfall model

Some advantages of waterfall model are:

1. It is easy to understand and use.
2. It is easy to manage due to strictness of the model.
3. Each step once completed and processed at a time.
4. Best for small projects.

Disadvantage of waterfall model are:

1. Not suitable for project where requirement are not defined properly.
2. It does not work better complex projects.
3. There is high amount of risk and uncertainty.
4. There is no going back after completion of a stage as it is very difficult.

## Design pattern

I have used MVC deign pattern for this project. It is the most popular framework for software development nowadays as it can be applicable for both web-based and desktop applications.

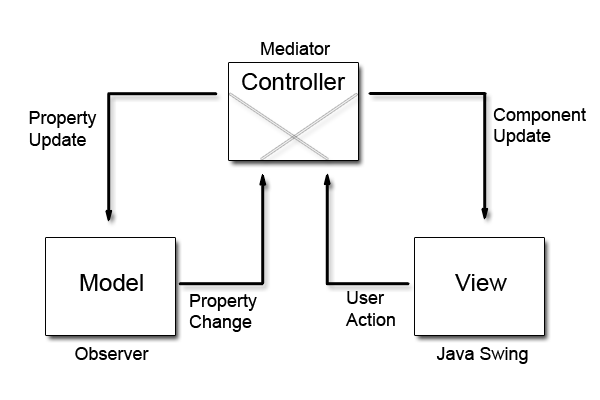


Figure 2: Design Pattern

Model: it is responsible for data that user works and represents the data between view and controller

View: it is responsible to handle UI of the application.

Controller: It acts as the main point between model and view as it handles the changes and requests.

## System architecture

The Arm system architectures define components and interfaces that make it easier for hardware and software to interoperate. (System Architectures, 2017)

## Project planning

## Work Breakdown Structure

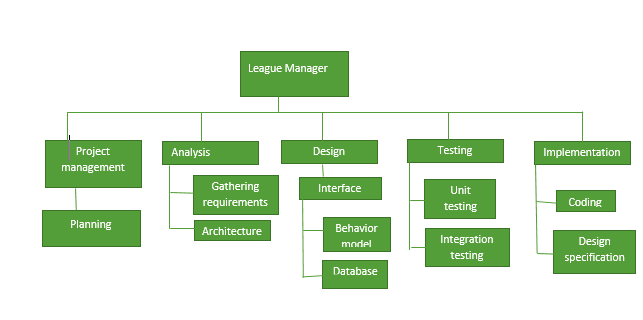


Figure 3: Work Breakdown Structure

Work break down structure is used to break down the project properly into manageable components. It helps us to assign responsibility, estimate cost, time risk, and difficulties. It also show the serial steps to be done while doing the project. In the above figure, I have shown how main task are divided into small tasks.

## Milestone

|  |  |  |
| --- | --- | --- |
| **WBS** | **Task name** | **Days** |
| 1 | **Project management** | **16 days** |
| 1.1 | Planning | 16 days |
| 2 | **Analysis** | **28 days** |
| 2.1 | Requirements | 20 days |
| 2.2 | Architecture | 8 days |
| 3 | **Design** | **25 days** |
| 3.1 | Database | 5 days |
| 3.2 | Interface design | 15 days |
| 3.3 | Behavioral design | 5 days |
| 4 | **Implementation** | **20 days** |
| 4.1 | Coding | 15 days |
| 4.2 | Design specification | 5 days |
| 5 | **Testing** | **7 days** |
| 5.1 | Integration testing | 4 days |
| 5.2 | Unit testing | 3 days |
| 6 | **Final Documentation** | **12 days** |
|  | **Total** | **108 days** |

### Figure 1 : time estimation

|  |  |  |
| --- | --- | --- |
| **S.N** | **Milestone** | **Date** |
| 1 | Proposal | 25th March, 2019 |
| 2 | Analysis | 10th April, 2019 |
| 3 | Design | 9th May, 2019 |
| 4 | Implementation | 4th June, 2019 |
| 5 | Testing | 25th June, 2019 |
| 6 | Documentation | 2nd July, 2019 |

Figure 5: Milestone

## Scheduling

A Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer and social scientist. (Rouse, 2007) The advantage of Gantt Chart is that it shows the proper progress of the project properly. It shows detailed information about the tasks. It can also show the information in a diagram form. The Gannt chart for my project is shown below:

## Time estimation

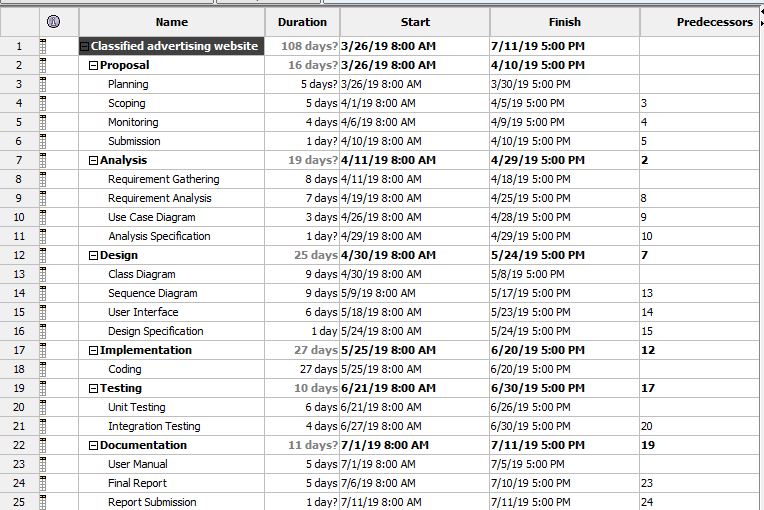


Figure 6: Time estimation

## Gantt Chart

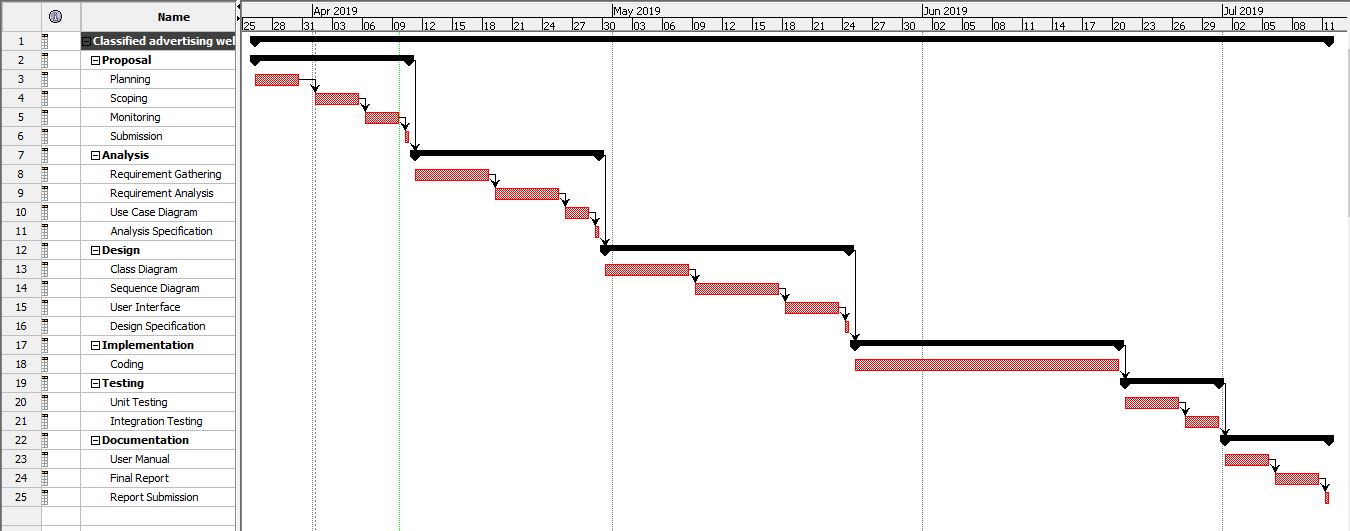


Figure 7: Gantt Chart

## Risk Management

Risk management means identifying the risk that may occur during the project, which can harm the productivity of the project and cause different obstacles that can affect our project. If not analyzed properly it we may not be able to complete our project on time. It is very important as it can help us prepare for the upcoming risk that occur in the project because risk can affect our project and solution for every risk should be there.

1. Identify the risk

This step shows or list out the risk that can occur during the project. All the possible risks should be listed. We can learn to how to deal with those type of problems or prevent ourselves from having that problem.

1. Analyze the risk

When all the risk are listed out and identified properly, we have to decide the likelihood and consequences it brings. We should understand the nature of the risk and how it affects the project.

1. Evaluate risk

In this process we determine the likelihood and consequences of the risk that results to the magnitude of the risk and then rank them. We have to decide if the risk is acceptable or not.

1. Treat the risk

In this step we determine how we can handle the highest ranked list and have a plan to stop those risks.

1. Monitor the risk

Here we see how the risk has affected the project by monitoring, tracking and reviewing the risk.

The table for likelihood and consequences of the risk we have created a table and given value as per their impact:

|  |  |
| --- | --- |
| **LIKELIHOOD** | **VALUE** |
| LOW | 1 |
| MEDIUM | 2 |
| HIGH | 3 |

|  |  |
| --- | --- |
| **CONSEQUENCES** | **VALUE** |
| VERY LOW | 1 |
| LOW | 2 |
| MEDIUM | 3 |
| HIGH | 4 |
| VERY HIGH | 5 |

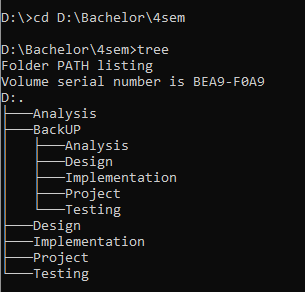
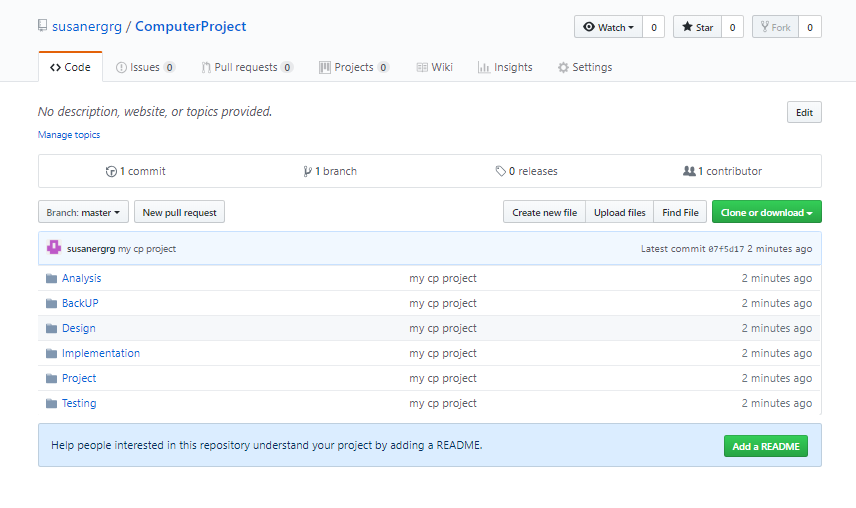
The impact is calculated as per the likelihood and consequences multiplied together:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N** | **Risk** | **Likelihood** | **Consequences** | **Impact** | **Action** |
|  | Crashing Of Hard Disk | 1 | 4 | 4 | Using online storage system. |
|  | Natural Disasters | 1 | 3 | 3 | Having different type of storage type. |
|  | Time management | 2 | 3 | 6 | Follow our schedule properly as shown in the gant chart. |
|  | Unqualified  employee | 2 | 3 | 6 | Providing proper training |
|  | Improper analysis | 1 | 5 | 5 | Spend time properly and wide thinking. |
|  | Lack of experience | 2 | 3 | 6 | When the project is started we should do it properly. |
|  | Lack of resources | 1 | 2 | 2 | We should have our resources ready before the project starts cause it may have stop the project. |
|  | Diseases | 2 | 3 | 6 | There should be healthy eating so that I can complete all project on time. |

# Configuration management

Configuration management means managing the configuration of all the project’s key products and assets. (Wrike, 2016) This includes the finished product that will be up for sale or delivered to the customer. Any changes that happen must be observed as well as measured to determine the impact on project. GitHub is used for storage of data as it can act as a regular backup as it can be used for recovery in case of some failures during the project.

<https://github.com/susanergrg/ComputerProject>



# Chapter 2: Analysis

Analysis is the first stage of software development either desktop or web where detailed examination is done. Analysis is done before the program is created, with the given scenario or the scenario of behavior of the program. It helps to determine requirement, the specified requirement by the user and finally checking the feasibility if the data that has been analyzed.

There are few points why analysis is needed:

1. Helps to know about future problem.
2. Potential upgrades in the program.
3. Fulfill all the requirements.
4. Requirements required (hardware, software).

There are many types of analysis process but for this project, I have chosen SWOT analysis, which means Strength, Weakness, Opportunity and Threat.



* Strength: It shows the strength of the company thing that company can do the best. Those qualities that our company can do better than the competitors as well as it includes the resources within like skill of the staff, capital, intellectual property etc.
* Weakness: It includes the limitations in our company that needs to be focused on such as limited resource.
* Opportunities: It means all those parts in which the company can shine and be at it best such as having less competitors in a certain area or in products.
* Threat: Threats refer to that problem that can harm the company. There are two types of threats :

External threat: These are those threats that can be affected by the outer means such as culture, trends, competitors etc.

Internal threat: These are those threats that can occur within the company such as low budget, less skillful, limited resources, etc.

# Feasibility study

After the analysis is done, it is not fixed if the software that has been developed will be a success or not and further can it satisfy the user or not. There are certain feasibility that can help to determine the feasibility of the program. They are:

1. Economic Feasibility: Considering the budget the application should be built. The expenses should be considered as the budget is limited and later either the sale of program can benefit the company or not.
2. Technical Feasibility: Only with proper technical supplies, we can completely build the program demanded by the user as per wish. In lack of any technology, the program may not be a perfect one.
3. Schedule Feasibility: We should work as per schedule made which should be done properly. In case of any issues, the project may not be completed in time.
4. Operational Feasibility: It refers to the skill of solving problems.
5. Legal Feasibility: It determines if the project developed is not against the law of the country or against ethical rules.

# Requirement analysis

## Functional

They refer to those requirements that have their own specific role in the project. It can change as per the projects. They are one of the most important function of the program and determine the application. It fulfills the requirement of the user or client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Function** | **Data Required** | **Dependency** | **Remarks** |
| F01 | Login | Username and Password | N/A | It helps to secure the system |
| F02 | Create a league | N/A | F01 | To start a league. |
| F03 | CRUD for teams | N/A | F02 | To organize the teams properly. |
| F04 | CRUD for players | Data of teams | F03 | To assign players as per their teams |
| F05 | Automated fixtures | Data of teams | F02 | Fixtures will be automatically scheduled. |
| F06 | Automated Table update | Fixture results | F05 | Tables updated after games |
| F07 | Data in games | Automated Fixtures | F05 | Data about games to be inserted |

## Non-functional

These features help to support the functional requirement or enhance the functional requirement. They are not important to be executed.

The non-functional requirement are:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Function** | **Purpose** | **Remarks** |
| F01 | Security | Secure the system | Increases security. |
| F02 | Efficient | Increase productivity | Helps to obtain best results. |
| F03 | Accessible | Easy to use | Friendly to use |
| F04 | Extensible | Further upgrades | Enhance features |
| F05 | Testable | Find solutions easily | Less error free and no bugs |

# MOSCOW Prioritization

It is one of the most popular technique used which helps to determine the important factor or the needs and separate which is not needed. It helps us to classify which is important and which is not so that we can allocate our time and effort.

The classification are:

1. Must have: these are the must have in the project that cannot be replaced or removed. They are the core of the program
2. Should have: these are those features that needs to be included but is not a must as without it the program can still run properly and needs to be included if possible.
3. Could have: They can be replaced or even removed as their presence doesn’t make a significant change in the program.
4. Won’t have: They are the requirements that should be removed from the project and don’t have any impact on the efficiency of the program.

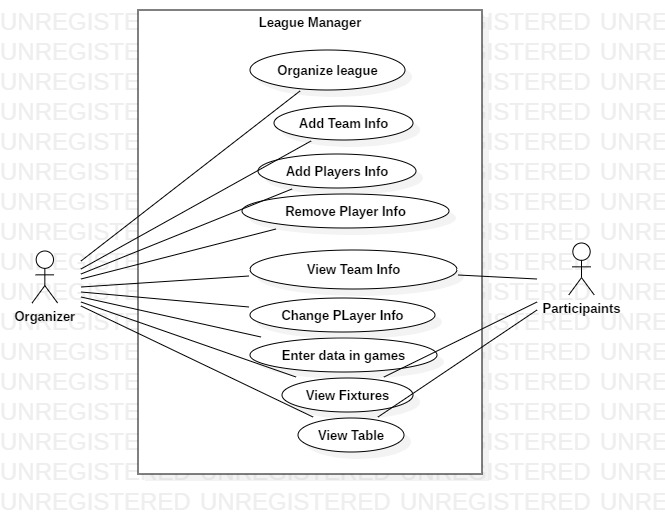
|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **Priority** |
| FR01 | Login | Must Have |
| FR02 | Create a league | Must Have |
| FR03 | CRUD for teams | Must Have |
| FR04 | CRUD for players | Must Have |
| FR05 | Automated fixtures | Should Have |
| FR06 | Automated Table update | Should Have |
| FR07 | Data in games | Must Have |

FUNCTIONAL MOSCOW TABLE

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **Priority** |
| FR01 | Security | Must Have |
| FR02 | Efficient | Must Have |
| FR03 | Accessible | Must Have |
| FR04 | Extensible | Should Have |
| FR05 | Testable | Must Have |

NON-FUNCTIONAL MOSCOW TABLE

# Use Case



Use case diagram are those diagram that shows the basic context of the program which is developed at the initial phase of software development. It shows the interaction of user in the program. The diagram consists of actor, use case and communication link.

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **Description** |
| 1 | Organize League | Only organizers can organize a league |
| 2 | Add team info | Teams are needed for a league |
| 3 | Add Player Info | Add details about players of the team |
| 4 | Remove player info | Remove player info that is mistaken |
| 5 | View team info | Everyone should be able to see the information of the team |
| 6 | Change Player info | Change information about players |
| 7 | Enter data in game | Events that happened in game |
| 8 | View fixtures | Everyone should see the information of the team. |
| 9 | View table | Everyone should see the table |

# NLA

NLA stands for Natural Language Analysis it is we classify the noun, verb and adjective from the given scenario. From the filtered process nouns are classes, verbs are method and adjective are attributes.

## Assuming Our Scenario

There will be a league organizer with a specified name and the teams can be added. There will be players in the teams that represent the team individually. The fixtures among the teams will be automatically created and the table will be updated as well. The events that happens in the team will be recorded by the admin only not by others. Participants can look at their fixtures, table as well as their info about the team. Whereas the organizer can have a lot of access such as CRUD for team and players.

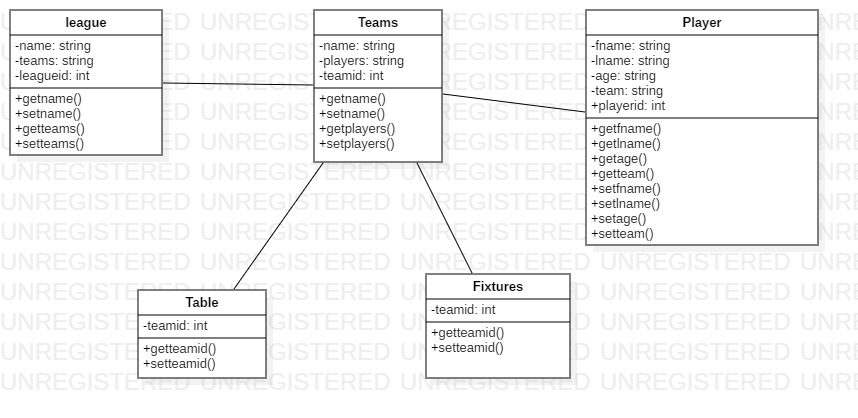
## Unfiltered Selection

Noun: league, admin, team, player, match table, event, fixtures, participants, record.

Verb: organize, specified, add, record, create, CRUD,result

## Filtered Selection

|  |  |
| --- | --- |
| **Class/Noun** | League, Team, Player, Fixtures, Table |
| **Method/Verb** | CRUD on teams/players, result, organize |



# Chapter 3. Design

## 3.1 Introduction

Design means transforming users need into suitable form, which helps the programmer in software coding and implementation. Users need is used for creating a design, as it must full all the requirements. It is the third phase of waterfall model after analysis.

Advantages of designing:

* Easy to use for the users.
* Fulfill the requirement.
* Good design leads to good relationship between customer and the software developer.

## 3.2 Structural Design

Structural design is the overview of the program, how the program will work and what are the requirements for it. It also helps in the architectural design,

There are many examples of structural design like class diagram, flowchart and data flow diagram. For this project, I have created class diagram and flowchart.

### Class diagram

Class diagram is the detailed view of the classes how it interacts with others classes and what methods are used in that class. It helps to understand the relation as well as help in coding.

**Justification:**

The class diagram I have, created shows the relation between the classes and it also shows the methods and datatypes used in each class, which can help me in coding with less errors.

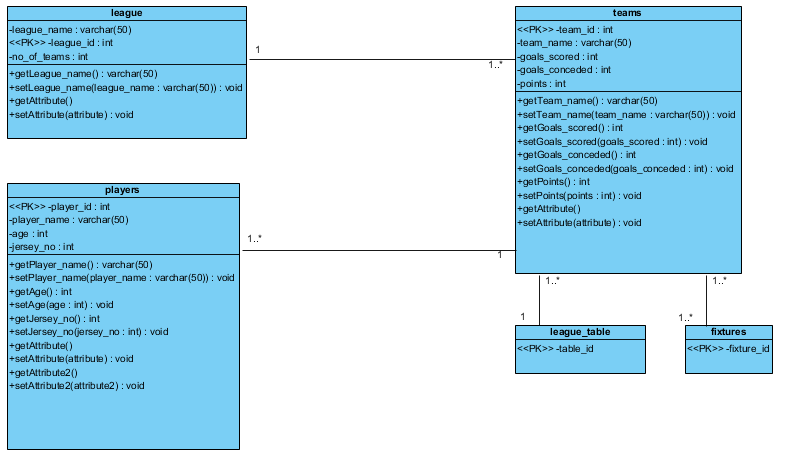
**Notation used:**

*Aggregation notation:* Used to configure objects for making complex type object.

*Composition notation:* Used for composite objects.

*Association notation:* Used to inter-relate class.

**Diagram:**

****

**Explanation:**

This is the relation between the entities used in my program.

### Data flow diagram (DFD)

Data flow diagram shows how the data flows in the system. It can also help us to determine the outputs as well as point where errors can take place in the program.

For this project, I have created DFD to know the detailed information how the data is going to flow.

**Justification:**

A data flow diagram can help a lot in the software development as it can overview how the data is flowing in the program which can also show the interaction between the classes.

**Notation used:**

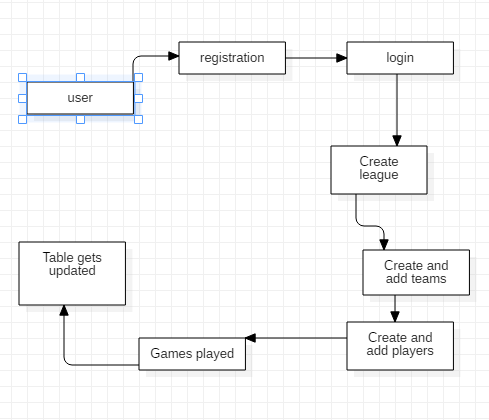
*Process Notation:* helps in data flow

*Data stores Notation:* repositories of the system.

*Data flow notation:* helps in determining flow of the data.

*External entity notations:* information about objects outside the system

**Diagram:**

****

**Explanation:**

This is how the data flows in the process and how it is executed properly.

## 3.3 Behavioral Diagram

Behavioral diagrams helps in visualizing, specifying, constructing, and documenting the dynamic aspect of the system. Behavioral diagram can be categorized as sequence diagrams, activity diagram, interaction diagram and state-chart diagram.

For this project, I have created use case and sequence diagram.

### Sequence diagram

Sequence diagram describes the interaction among classes in terms of an exchange of messages overtime. They are also called event diagrams.

I have created sequence diagram to have overview how the objects are interacting with each other and how they deal with each other.

**Justification:**

I have created sequence diagram to show the interaction between the objects and show how the data flows. It can clarify the users about how the program is going to work out.

**Notation used:**

*Lifeline:* This notation helps to represent the all instance in every interaction.

*Message:* This notation help to send the text.

*Message reply:* This notation help to reply the text to the object.

*Self-message:* This notation helps to write the text to own self-object.

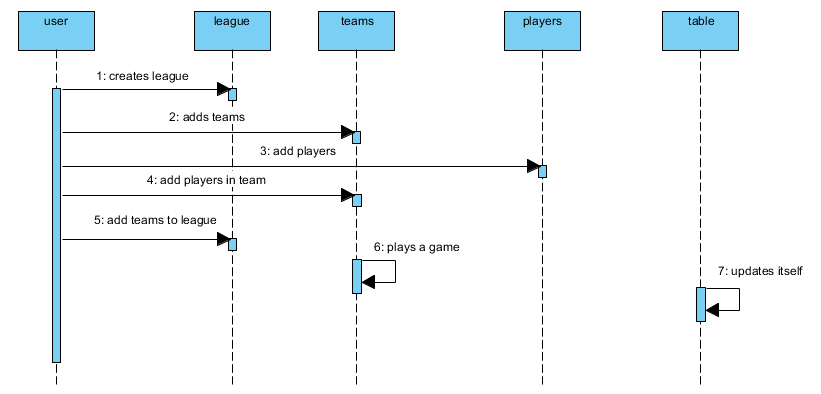
*Actor:* This notation is for the object, which is important for the system to run.

*Combine fragment:* This type of notation help to write down the if/else statement.

*Object:* This is the class like structure without which system flow is not possible.

This notation helps to represent the all instance in every interaction.

**Diagram:**

****

**Explanation:**

This is how the whole process is carried out and how the league is organized properly.

### Activity diagram

Activity diagram is the process of showing how a task is carried out by process in their specified classes. They show how the task are operated.

For this project I have created activity diagram and shown a full process of a task done.

**Justification:**

With the help of activity diagram, we can have more detailed information about how a task is carried out.

**Notation used:**

*Start Point:* This is the point which represents the initial action state.

*Action State:* This point represents the non interruptible actions of the step.

*Action flow:* This notation helps to flow the object from one action state to another.

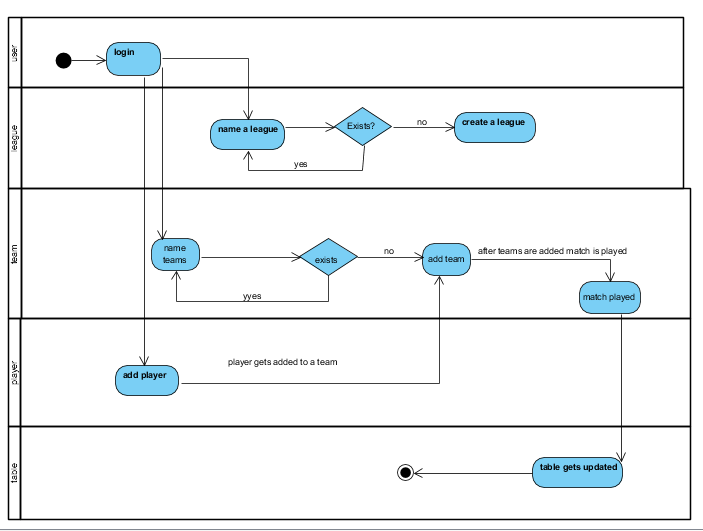
*Decision branching:* This notation helps to decide like if/else condition.

***Synchronization:***

*Fork node:* This notations help to split the one action state into two or many.

*Join node:* This notations helps to combine the two or many action state to one action state.

**Diagram:**

****

**Explanation:**

This is how the full process is carried out in the system.

## 3.4 Database management system

It is the system responsible for the creation of database as well as manage the software. It also helps us to perform CRUD operations. There are advantages of using DBMS :

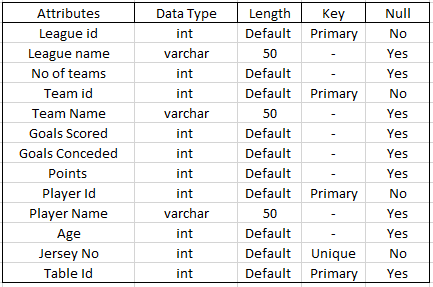
1. It helps to control redundancy.
2. Avoid inconsistency.
3. Integrity can be enforced.
4. More secure

### Data Dictionary:

Data dictionary is the file that contains the data of the data or metadata. It helps to know the type, length and many more information about the data.

**Justification:**

I have created data dictionary to avoid further problems that may occour during the coding.



### ER (Entity relationship):

Entity relationship gives the knowledge about the relation between entities.

**Justification:**

I have five entities for this project they are:

1. League
2. Teams
3. Players
4. Table
5. Fixtures

**Notation Used:**

*Fields:* This notation is used to classify the entity and their columns.

***Keys***

*Primary key:* This key uniquely identifies the column. It is unique as well.

*Foreign key:* This key are created any time an attribute relates to another entity that has same character from the related entity.

***Cardinality and Cordiality***

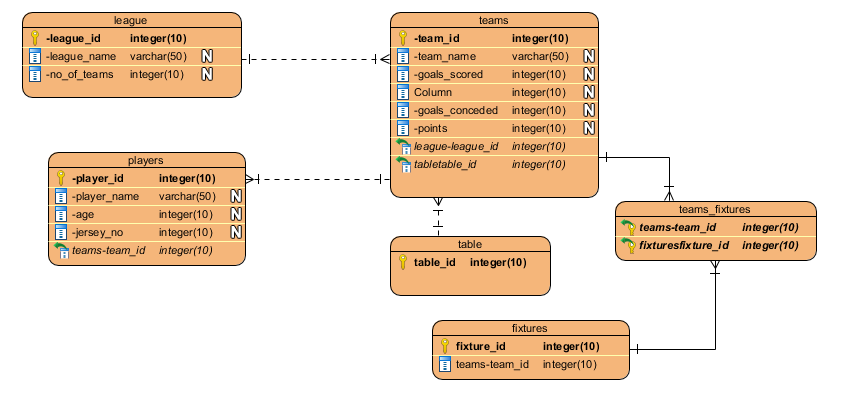
Cardinality and Cordiality are the notations that are used to relate the one instance of one entity to another instance of another entity.

There are 3 types of notation

*One to many:* This notation helps to relate the one instance of one entity with the many instance of another entity.

*Many to many:* This notation helps to relate the maximum number of times an instance in one entity with other many instance of another entity.

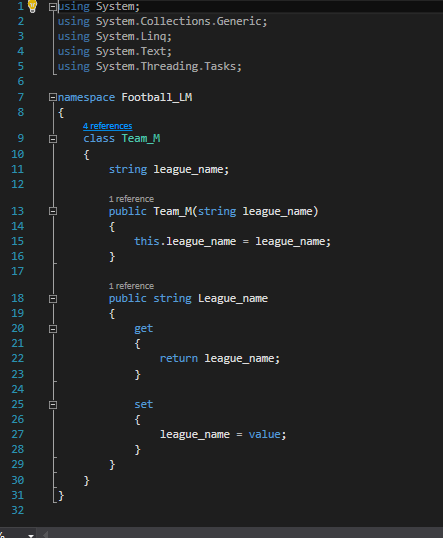
*Many to one:* This notation helps to relate the maximum number of times an instance in one entity with one instance of other entity.

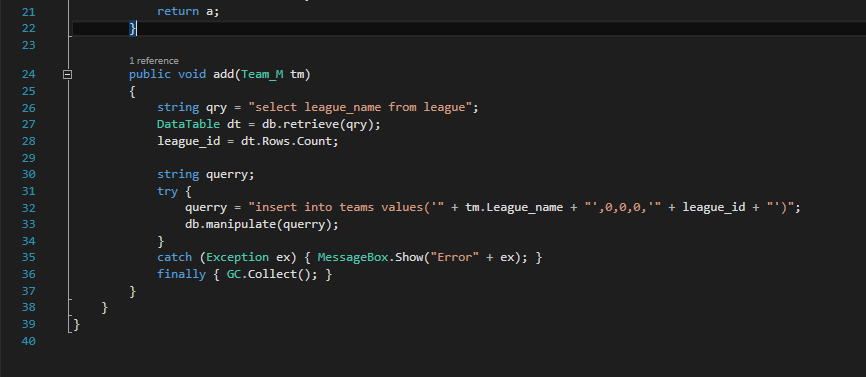


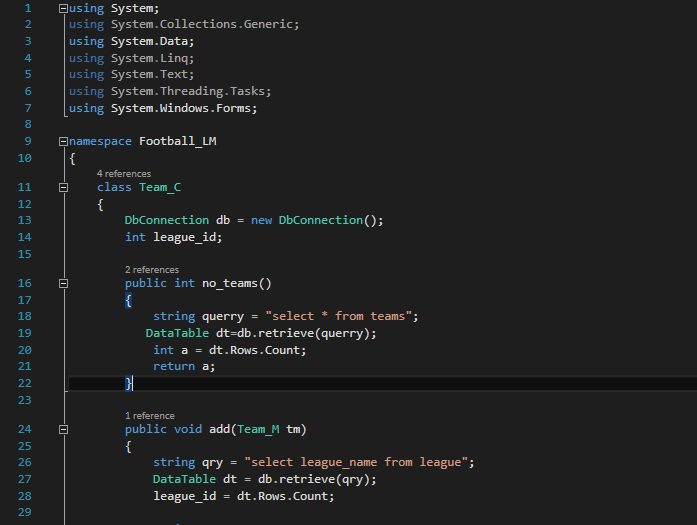
## 3.5 Architecture

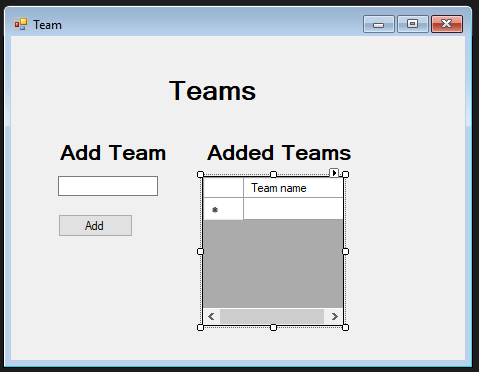
It is the fundamental structure that must be made to give shape to the system. It helps to design the system, edit if necessary and proceed to develop the system.

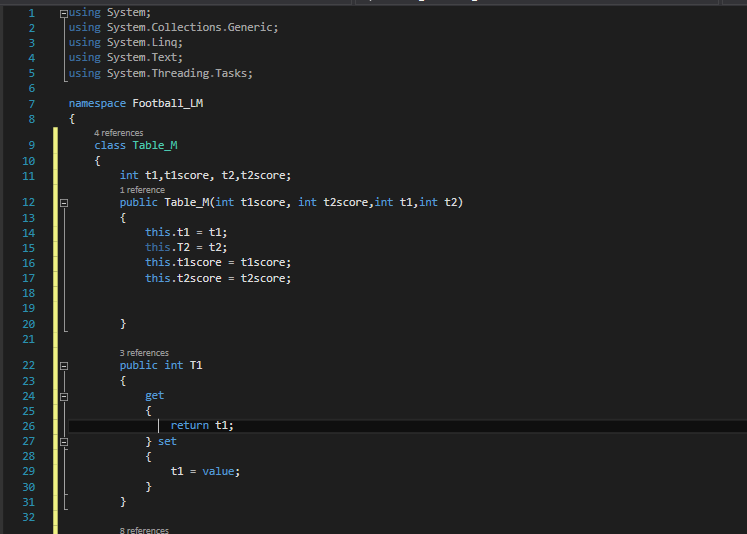
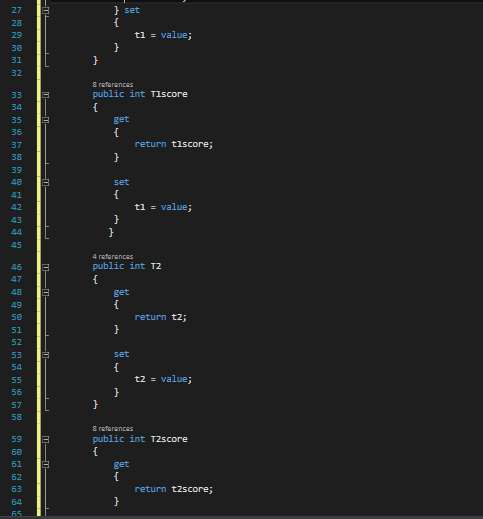
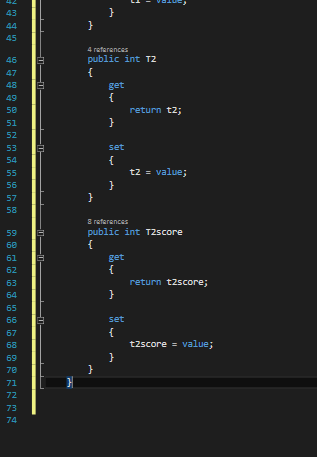
# Chapter 4: Implementation

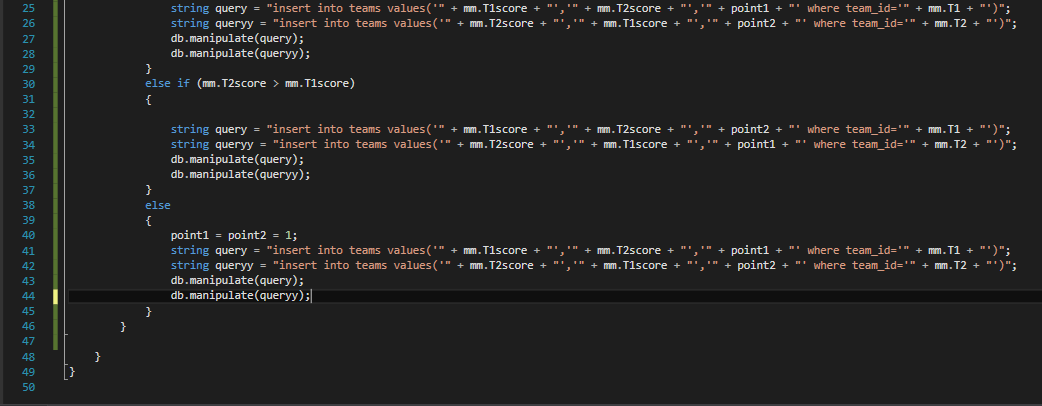




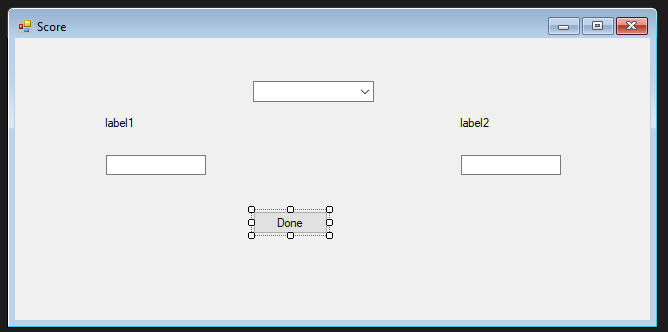


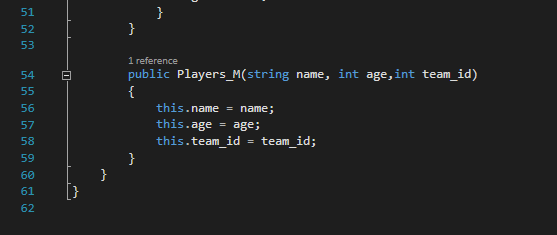


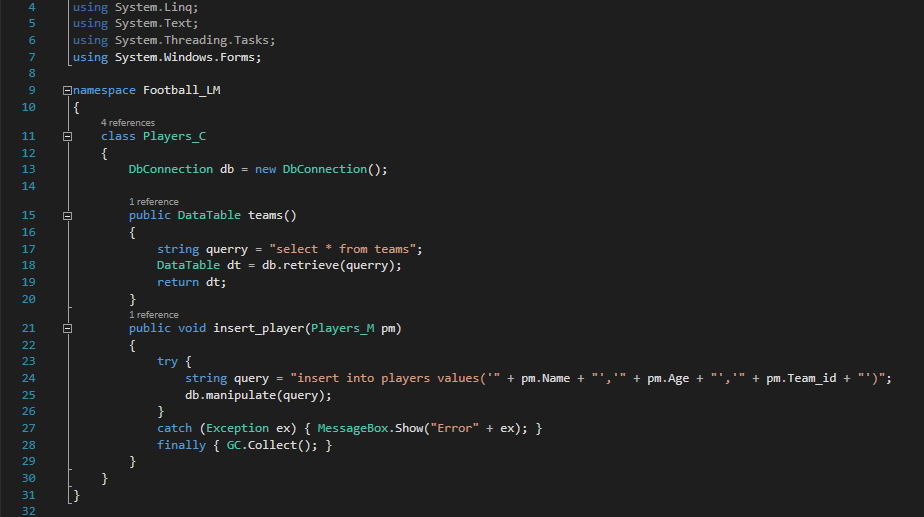
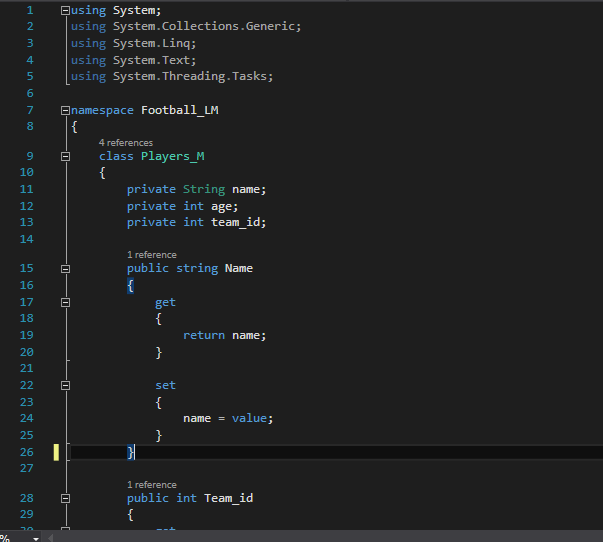
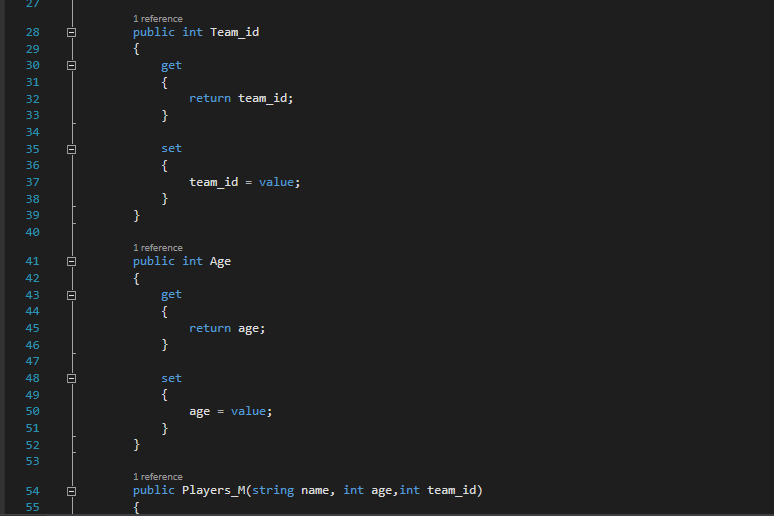


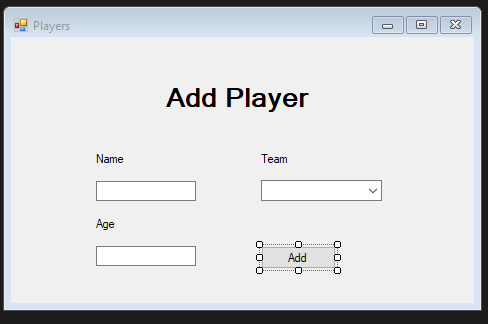


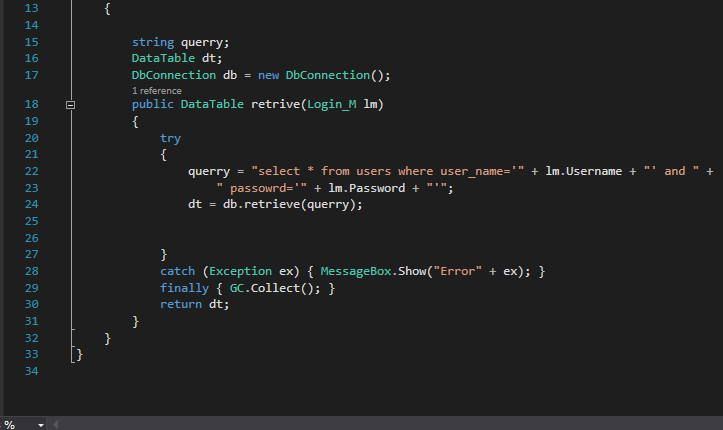
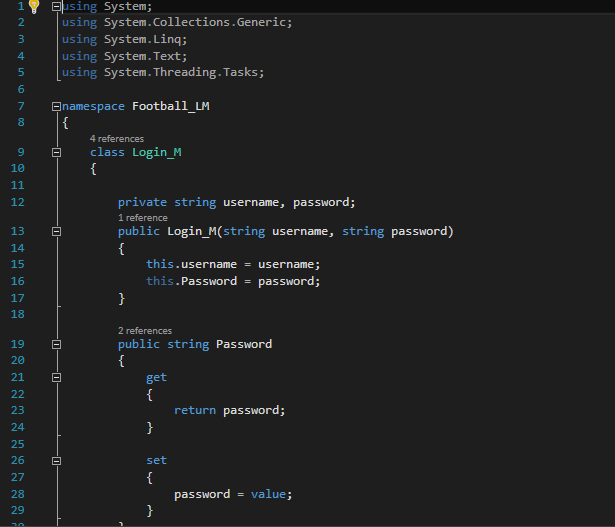
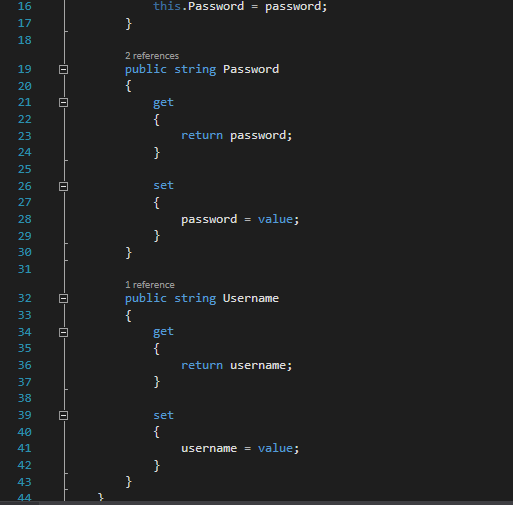


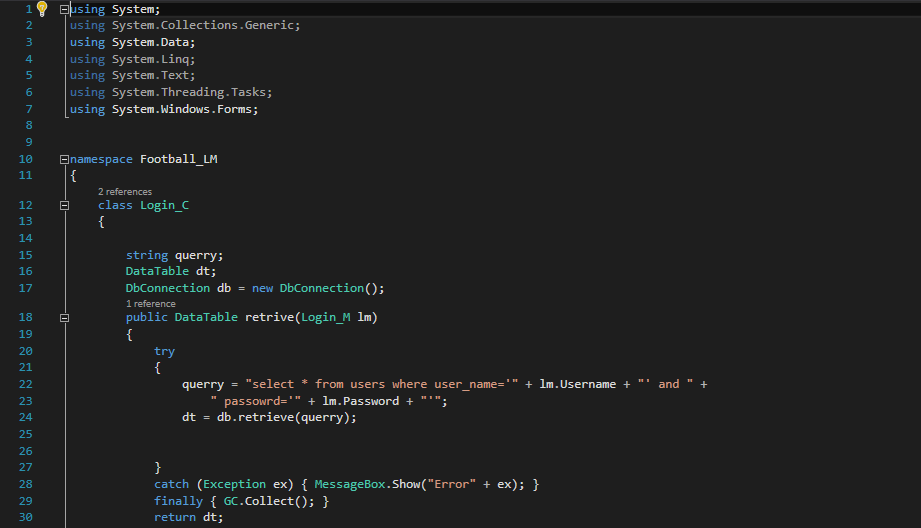


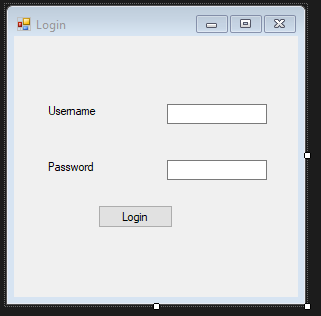


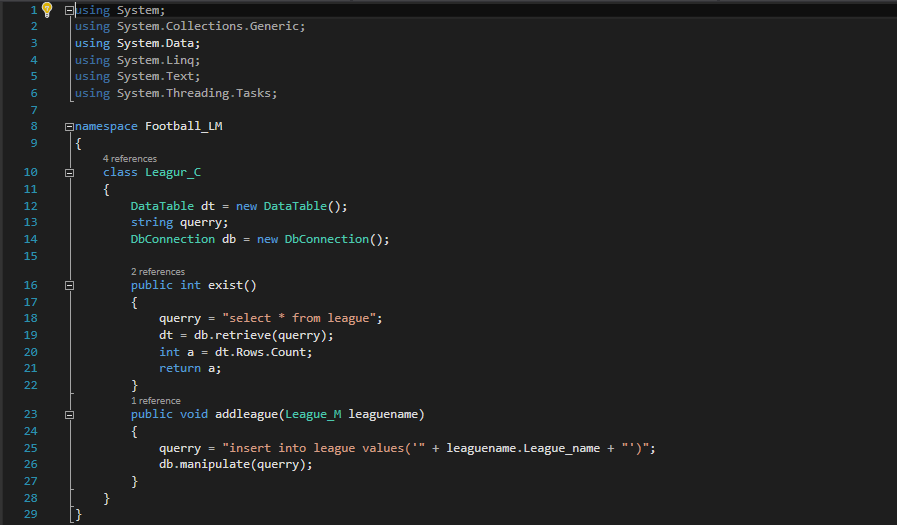


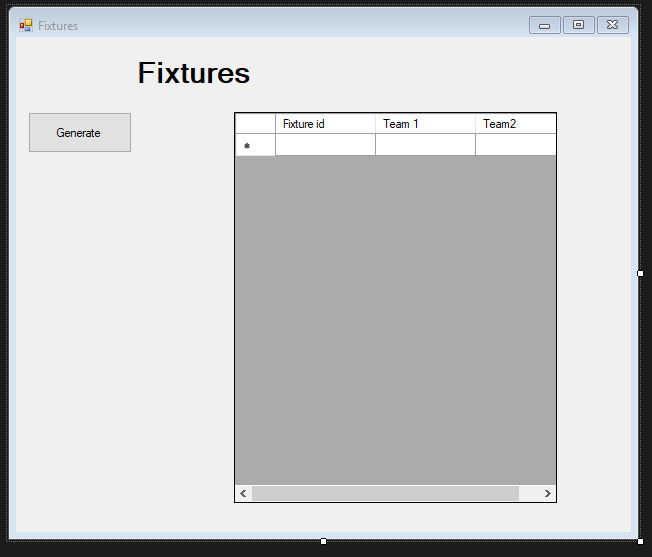
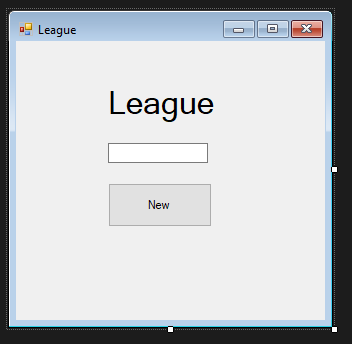
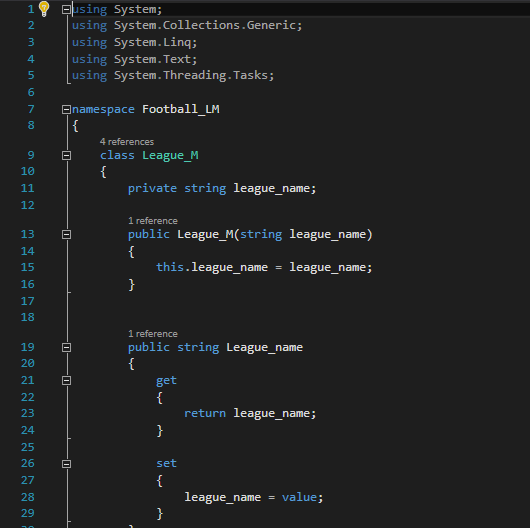
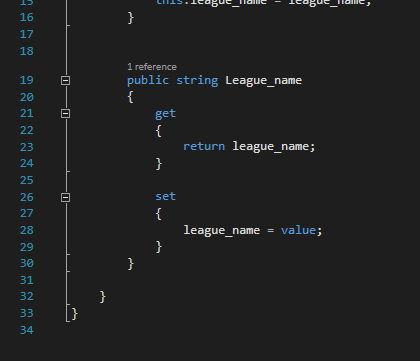


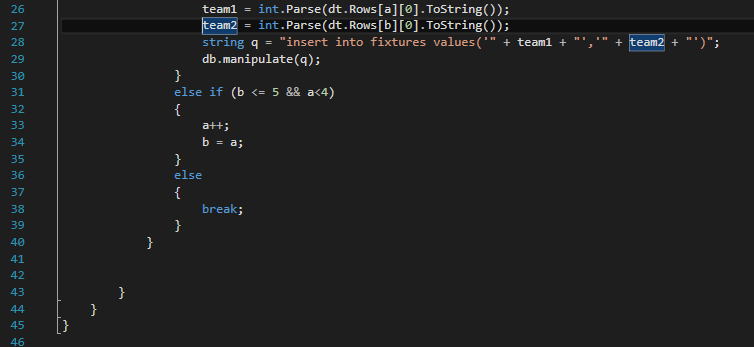


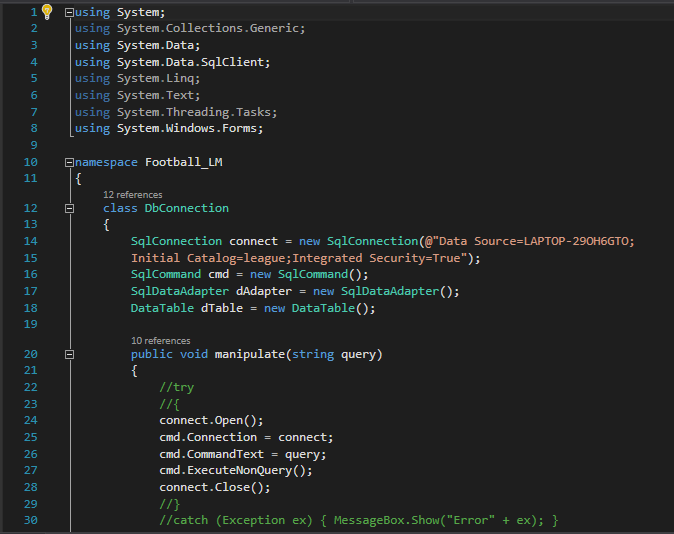
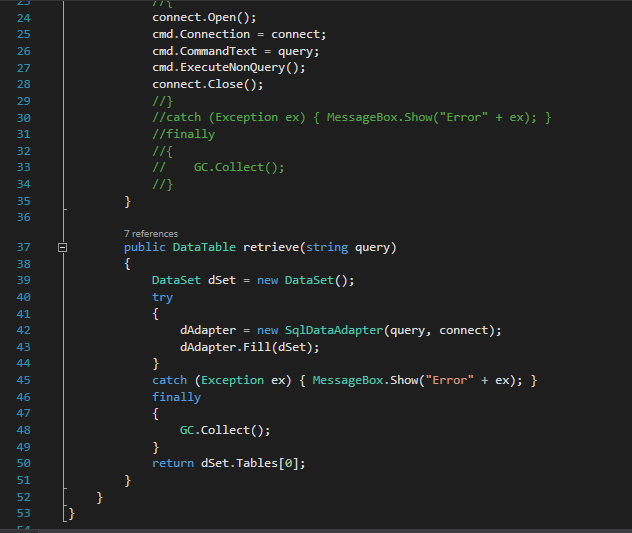
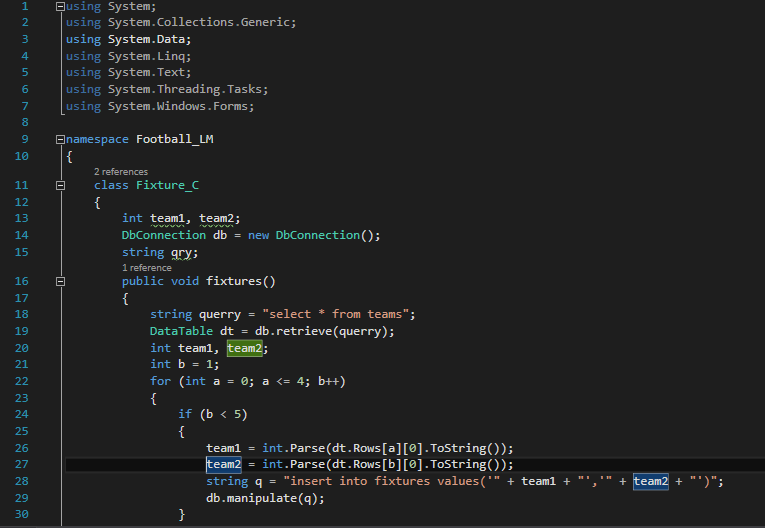


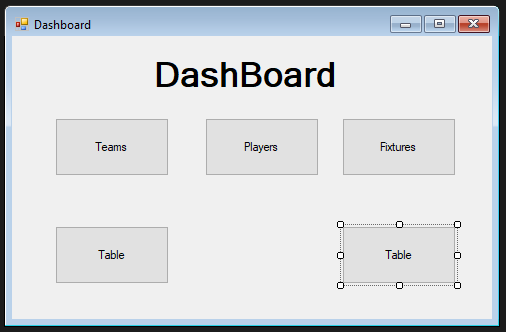












# Chapter 6: Issues

Other project issues includes configuration management, risk management, scheduling for future work etc. Configuration management helps us to configure changes whenever it’s necessary. One of the risk in this project is the team management sometimes team may vary and sometimes they may be absent in order to solve this problem proper team management is required. Secluding time for the game helps to manage time. This helps to reduce the amount of work required for managing the match. This time can be used for other works. The limitation of this application is that the space storing the matches can exceed it one time so to overcome this limitation database should be checked monthly and if it requires more space then we need to add the necessary space required for it. The project is simple and easy to use. Even if the staffs don’t know anything about it may only require 4 to 5 days of training. After that they can use the application with ease.

# Chapter 7:Conclusion

In the end of the project I have finally built a fully functional software. I had many obstacles on the way during the developing phase but with the help of my friends and teacher I was able to build the confidence to develop the project properly. I was did some research on the internet which was a big help during the implantation and testing phase.