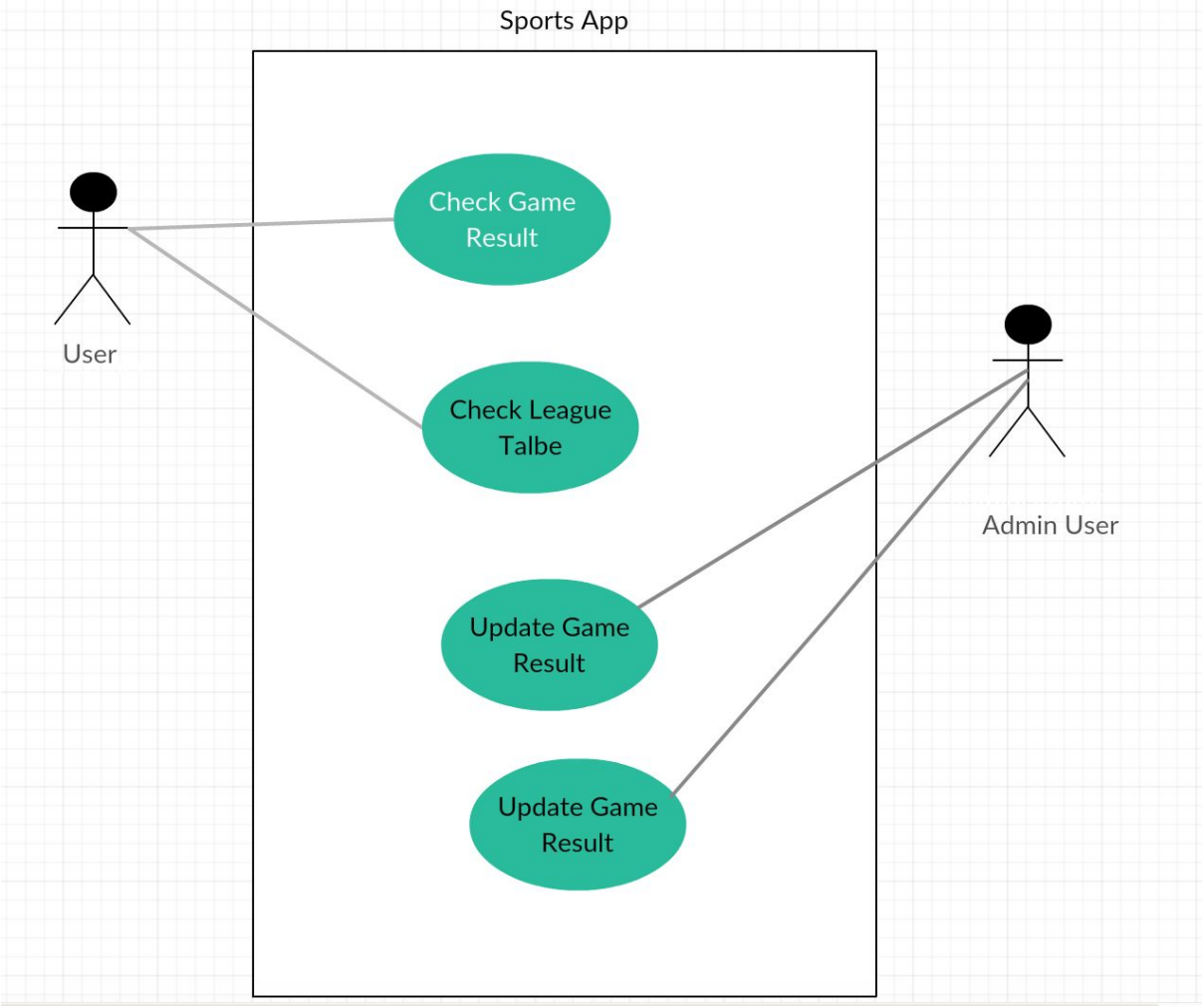
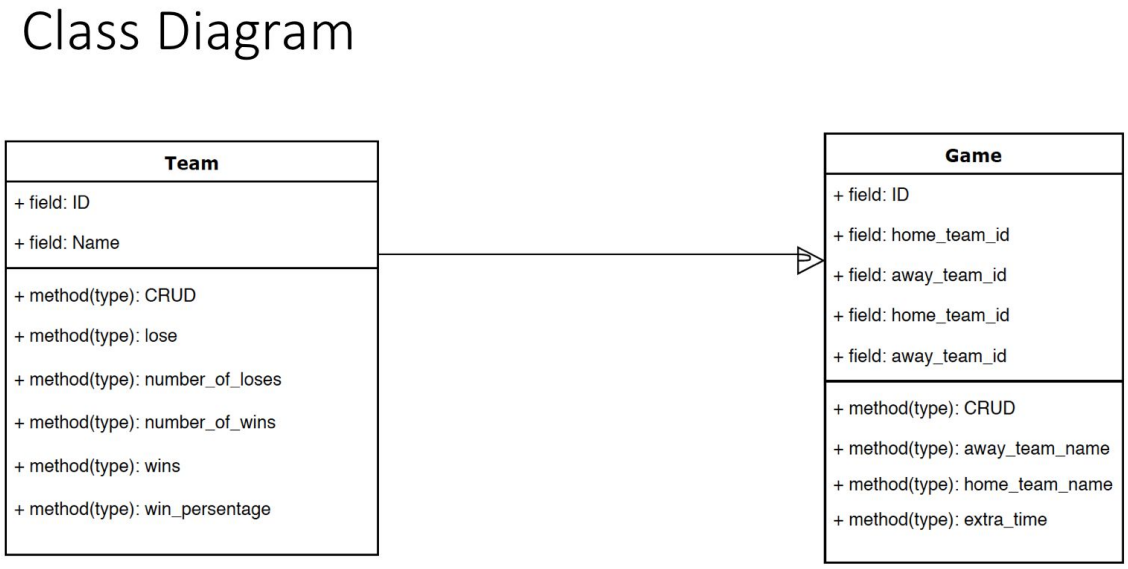
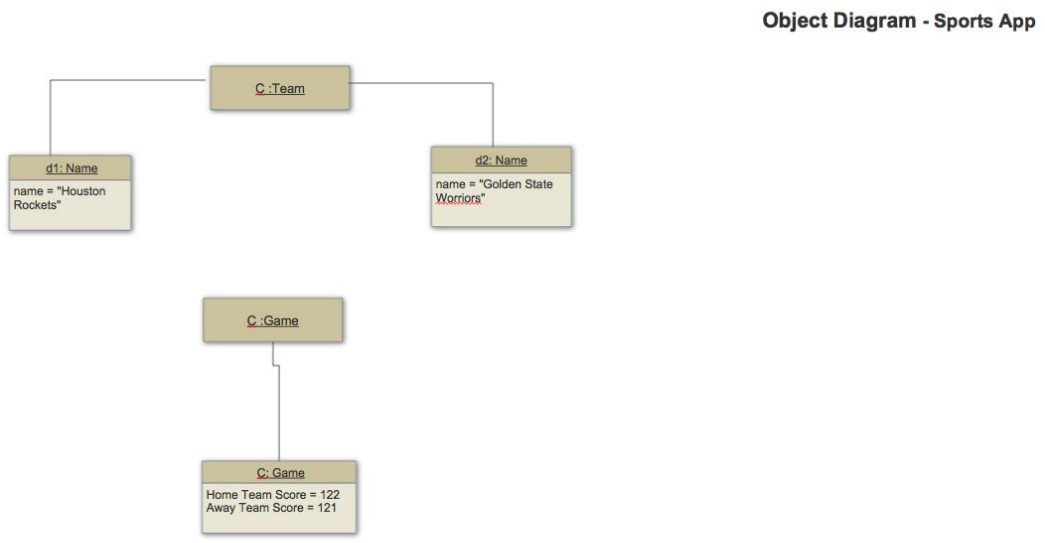
Week 5 A.D 1 A User Case Diagram



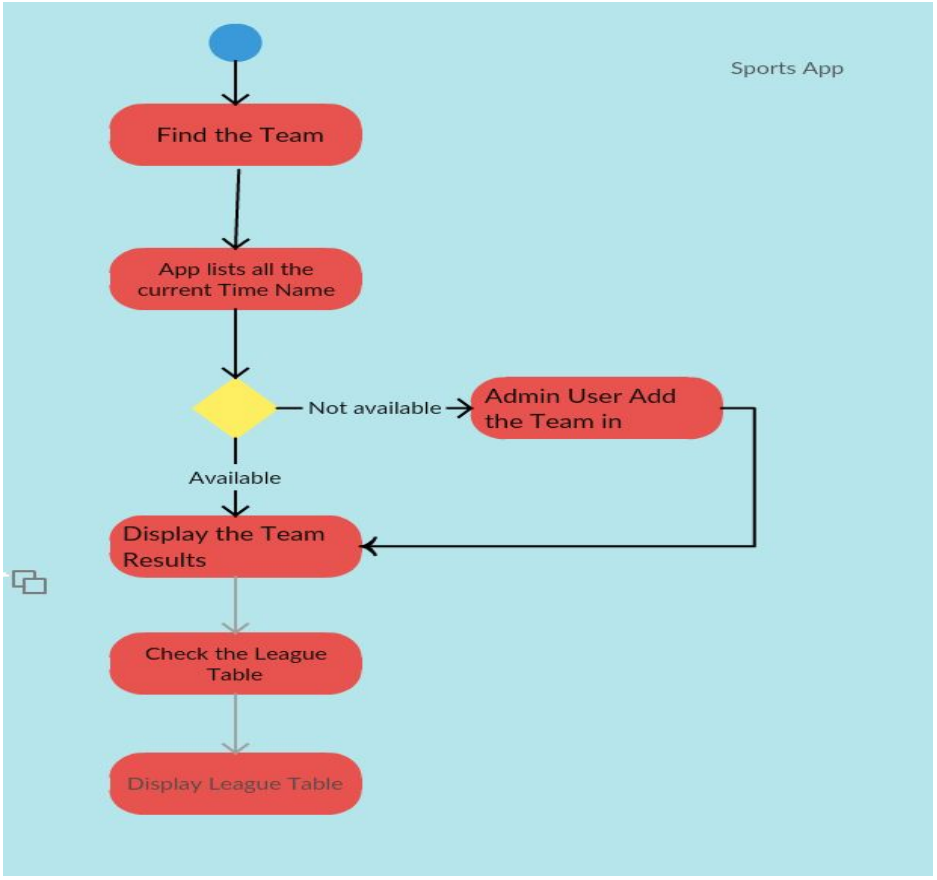
A.D 2 A Class Diagram



A.D 3 An Object Diagram



A.D 4 An Activity Diagram



A.D 6 An implementation constraint plan

Hardware requirements

The following table lists the minimum and recommended hardware requirements for running the sports app.

Component	Minimum	Recommended
Processor	2.5 gigahertz (GHz)	Dual processors that are each 3 GHz or faster
RAM	1 gigabyte (GB)	2 GB
Disk	NTFS file system–formatted partition with a minimum of 3 GB of free space	NTFS file system–formatted partition with 3 GB of free space plus adequate free space for your Web sites
Display	1024 × 768	1024 × 768 or higher resolution monitor
Network	Stand Internet Connection	Stand Internet Connection or faster

Software requirements

The Sports app is designed to fit to display on all type of browsers, it is recommended to browser it on Google Chrome.

Performance requirements

The Sports app is built on Ruby/Sinatra/SQL, so it would require any user to install Ruby/Sinatra/SQL before it can be runned.

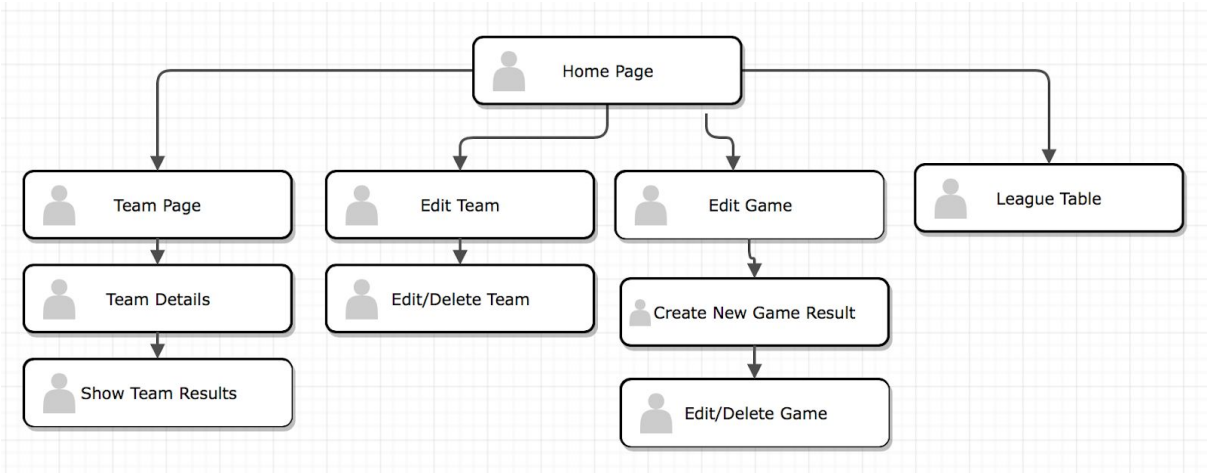
Persistent Storage and Transaction requirements

The Sports app is built with the aim of taking limited storage capacity, it should not require more than 5Mb of Hard Disk Space. The same apply to Transaction too.

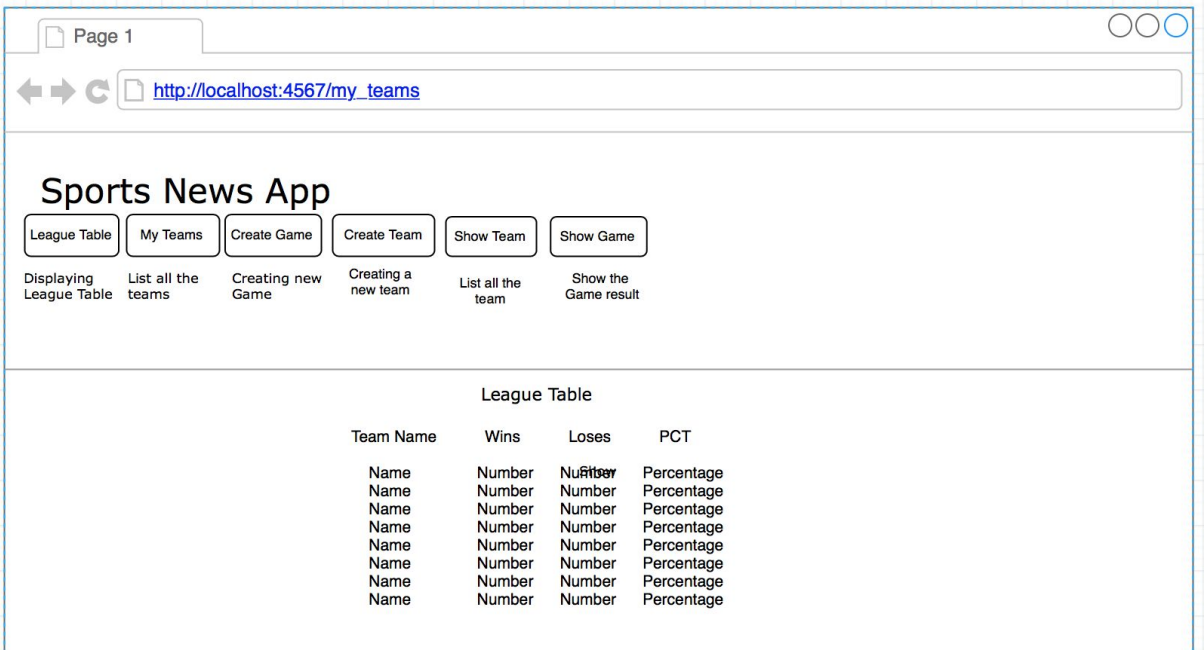
Budgets and Time

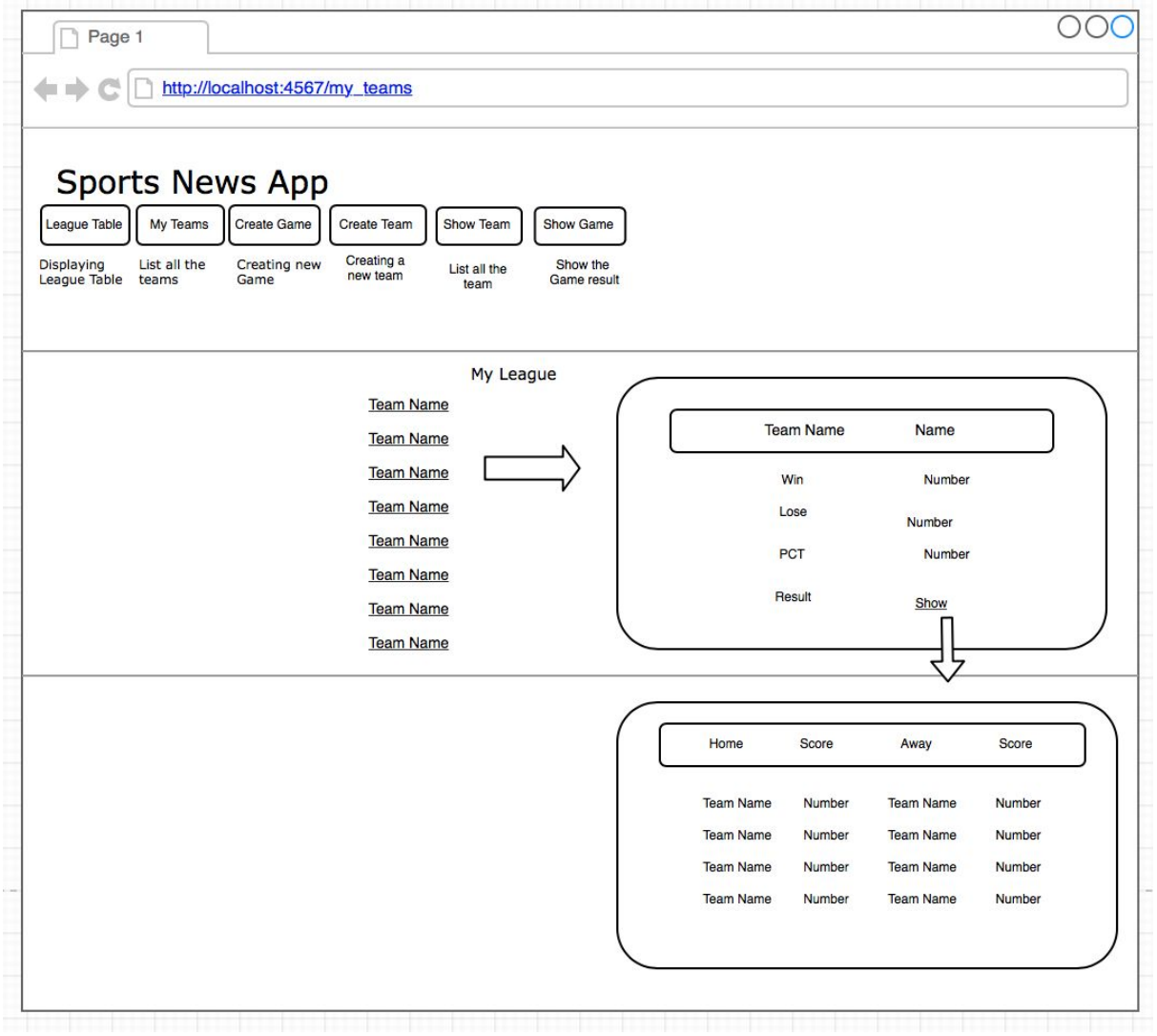
The Sports app was completed within 7 days, and there was no budgets set up at the initial stage, there could be a few interesting extension adding onto the project in the future, at this stage there has not been a future budget set either.

P 5 a User Sitemap



P 6 Two Wireframe Diagrams




Page 1
http://localhost:4567/my_teams
Sports News App
League Table
My Teams
Create Game
Create Team
Show Team
Show Game
Displaying League Table
List all the teams
Creating new Game
Creating a new team
List all the team
Show the Game result
My League
Team Name
Team Name
Team Name
Team Name
Team Name
Team Name
Team Name
Team Name
Team Name
Name
Win
Number
Lose
Number
PCT
Number
Result
Show
Home
Score
Away
Score
Team Name
Number
Team Name
Number
Team Name
Number
Team Name
Number
Team Name
Number
Team Name
Number
Team Name
Number

P10 pseudocode

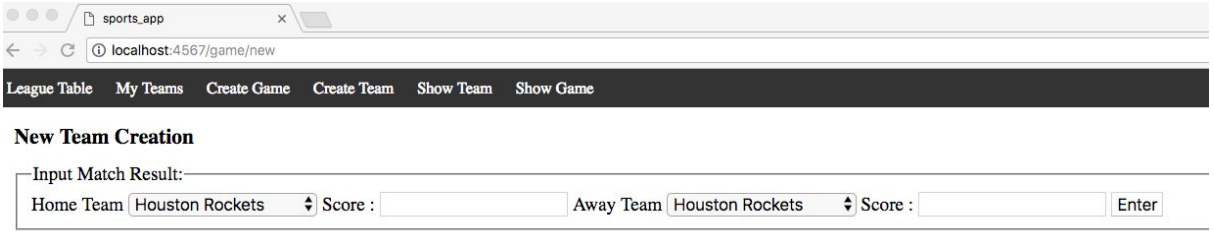
```
creating static method find all

method find all
select all games from database
save all the games objects in the result varialbe
pass individual games objests element to games variable
return games
method finish
```

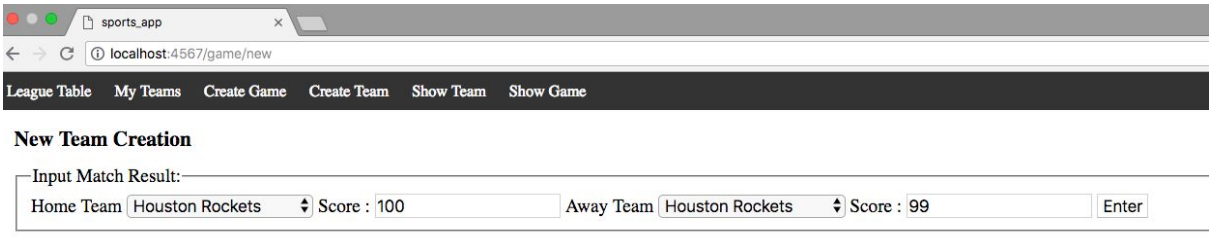
```
def self.all
  sql = "SELECT * FROM games"
  result = SqlRunner.run(sql)
  games = self.map_items(result)
  return games
end
```

P13 User Input

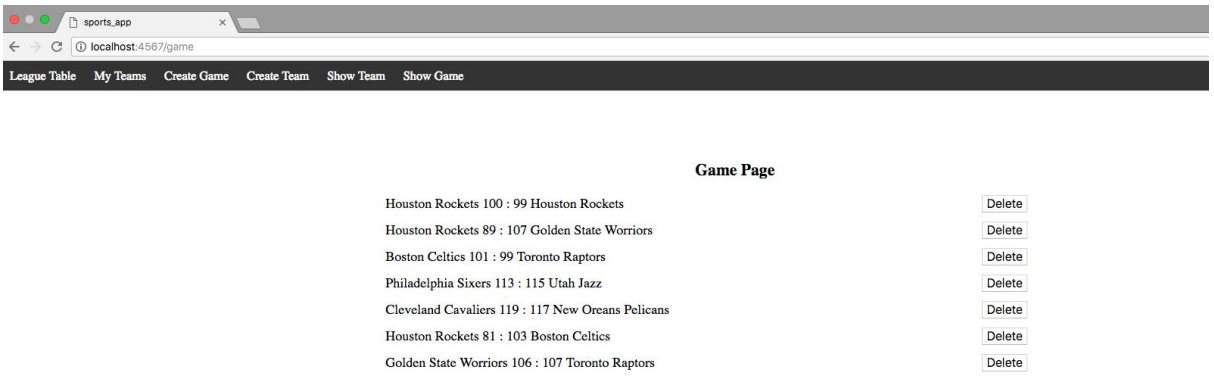
Step 1 User can input new Team result



Step 2 System take input from user



Step 3 System accept User Input, and System display your input



P14 Show an interaction with Data persistence

Step 1 Create New Team

Step 2 Added the Team to the league

P 15 show the correct output of results and feedback to user

Step 1 Click Edit Team Name

Step 2 Edit Team Name

Step 3 Name Changed