San Jose State University

Department of Applied Science

One Washington Square, 95112

DATA 225: Sec-22 Database Systems for Analytics

An Application for Interactive Query and Exploration of an NBA Dataset

INTRODUCTION

The National Basketball Association (NBA) is a professional basketball league in North America composed of 30 teams (29 in the United States and 1 in Canada). It is one of the major professional sports leagues in the United States and Canada and is considered the premier professional basketball league in the world.

Key functionalities of the application include the following:

Viewing: The application allows users to view data within the database, generating reports on players, teams, games statistics & player statistics. This basic exploratory functionality tests the database's completeness and accuracy.

Searching: Search features utilize the database schema's attributes as filters, enabling targeted retrieval of records based on criteria such as player name, team, date range, and statistical thresholds.

Filtering: The application provides custom filtering of database tables to isolate subsets of records for focused analysis. This taps the full flexibility of the schema's multiple granularities.

Updating: Authorized application users can edit database fields to correct errors and append new information, testing the integrity constraints implemented within the schema design.

Analyzing: Visualizations and customized queries generated through the application interface leverage the full analytical power of the integrated, longitudinal NBA dataset. Emerging trends may then be identified.

DATA SOURCE

The dataset utilized in this project was sourced from the Kaggle website (link here), where NBA stats website data was gathered, including game data, team stats, and player stats. The objective of this project is to create a user-friendly GUI application that allows individuals to access NBA data stored in MySQL.

The data source contains five files which are mentioned below:

- games_details.csv: details of games dataset, all statistics of players for a given game.
- games.csv: all games from 2004 season to last update with the date, teams, and some details like number of points, etc.
- players.csv: players details (name).
- ranking.csv: ranking of NBA given a day (split into west and east on CONFERENCE column
- teams.csv: all teams of NBA.

Operational Database Design

The application is mainly divided into two parts, Operational and Analytical. Operational covers the first four functionalities mentioned above, namely Viewing, Searching, Filtering and Updating. Analytical part covers Analyzing the data and producing graphs.

The relationship between each entity can be described by ER diagram which is a relationship diagram that explains 1:1 (ONE - ONE), 1:M (ONE - MANY), M: M (MANY - MANY)

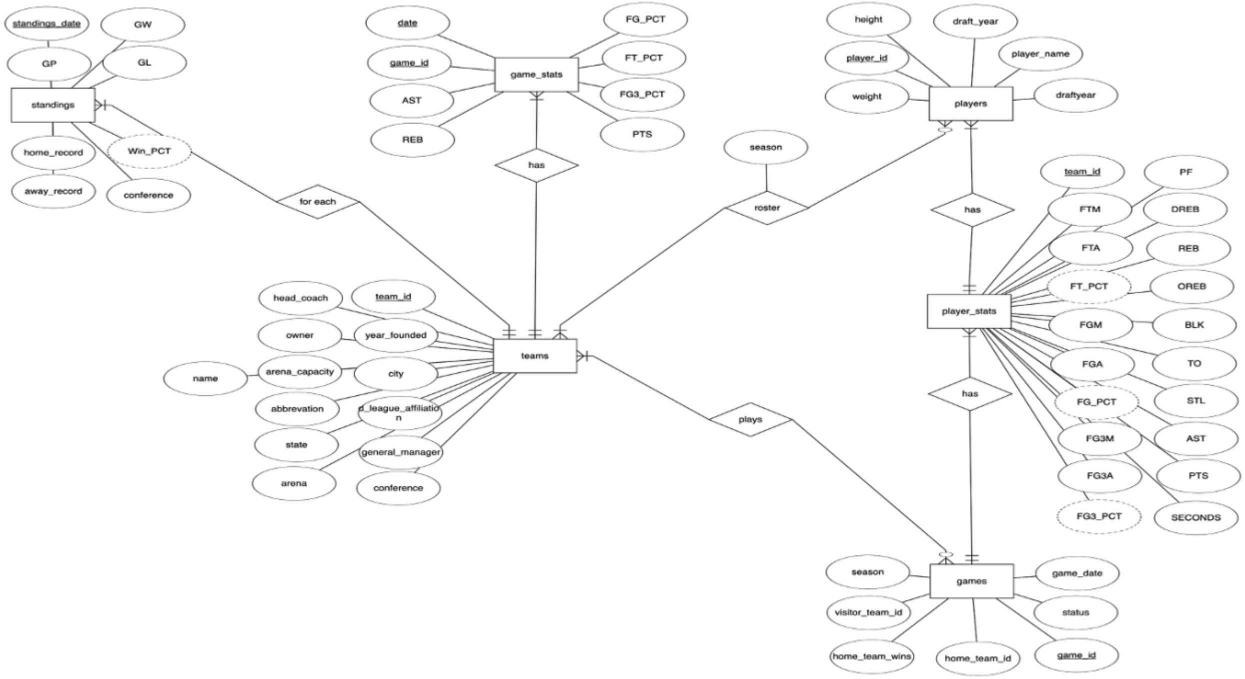


Fig. 1 ER Diagram of the Operational Database

The csv tables have been arranged in an effective manner into 6 entities like **standings**, **games**, **teams**, **team_stats**, **players**, **player_stats** and one junction table **roster** which is the result of teams and players entity having a M:M relation.

The entities and their attributes are mentioned below.

players - Stores information on individual players like name, height, weight, position, team, jersey number, and draft information. This allows you to query details about any current or former NBA player.

games - Contains information on each NBA game played like date, home team, away team, final score, venue, and season. This allows you to look up details about any game in NBA history.

teams - Stores information on each NBA team like name, city, arena, coach, general manager, and roster. This allows you to get details on any team, current or historic.

team_stats - Tracks team statistics over time like points per game, field goal percentage, rebounds, assists, and more. This allows you to analyze team offensive and defensive trends.

Player_stats - Contains statistics for each player in every game like points, rebounds, assists, steals, blocks, fouls, turnovers, and field goal percentage. This allows you to track player performances over their careers.

standings - Stores the standings, divisional rankings, wins, losses, and home/away records for each season. This allows you to see how teams fared and which teams made the playoffs for any given year.

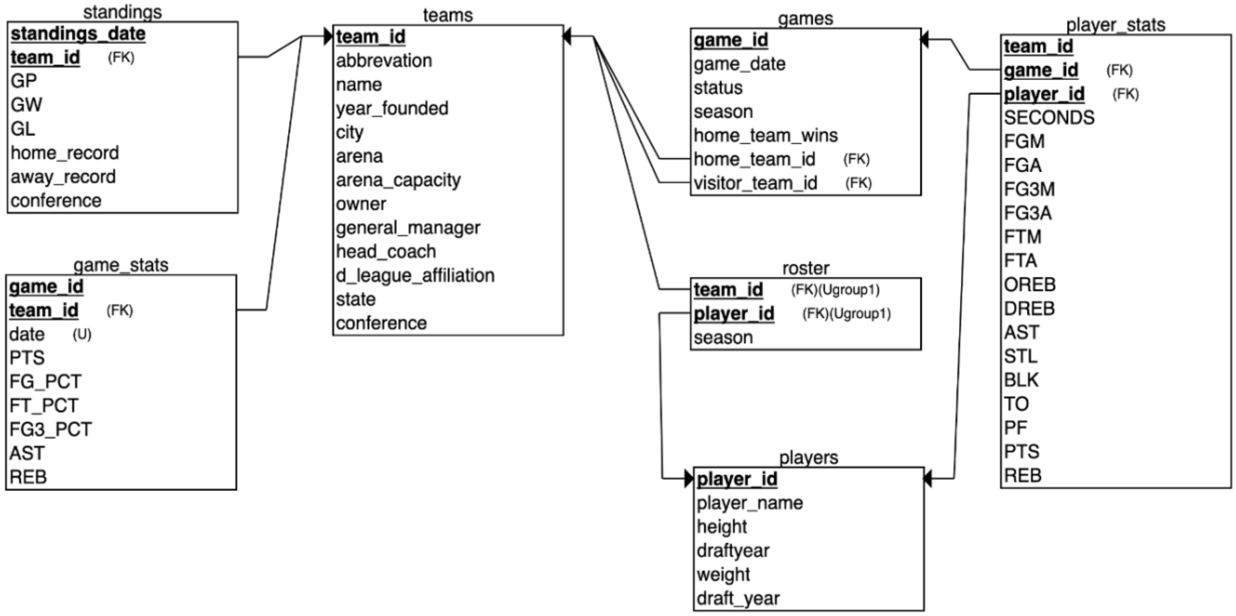


Fig 2. Relational Schema of the Operational Schema

ETL to Operational Database

There are few data inconsistency to be taken care of before loading the data into the database. One of which is missing data. The code below shows the missing players data for which there is games details but no player data.

```
df_all = games_details.merge(players.drop_duplicates(),
on=['PLAYER_ID'], how='left', indicator=True)
data = df_all[df_all['_merge'] == 'left_only']["PLAYER_ID"]
missingPlayers = pd.DataFrame(data = data)
```

To retrieve the missing data we have used the nba_api. We have particular used the CommonPlayerInfo method from nba_api.stats.endpoints and created the getMissingPlayerInfo function which takes player_id from the game_details.csv and gets the players data.

```
from nba_api.stats.endpoints import CommonPlayerInfo

def getMissingPlayerInfo(player_id):
returnCommonPlayerInfo(player_id).get_data_frames()[0][['FIRST_NAME','
LAST_NAME', 'TEAM_ID', 'HEIGHT', 'DRAFT_YEAR']]
```

The missing data in games_details.csv just means that the player has not played in that match. For the missing data in games.csv there is no data even in the official NBA site, as they are all concentrated in 2003 we will just drop 2003 data and start from 2004. The same will be done for ranking.csv file.

```
games = games[games.GAME_DATE_EST>'2004-01-01']
ranking = ranking[ranking.STANDINGSDATE>'2004-01-01']
```

In players.csv we decided to add more features like height, weight, birthdate and draftyear we used CommonPlayerInfo again to get the details for all the players.

```
from nba_api.stats.endpoints import commonplayerinfo

def getPlayerInfo(player_id):
    return
commonplayerinfo.CommonPlayerInfo(player_id).get_data_frames()[0][['CO
UNTRY', 'HEIGHT','WEIGHT', 'DRAFT_YEAR']]
```

Converting the height data which is an object data type to integer and converting foot data into cms.

```
def convertHeight(heightInFoot):
    if len(str(heightInFoot).split("-")) == 2:
        return(float(heightInFoot.split("
")[0]+"."+player.HEIGHT.split("-")[1]))*30.48
    else:
        return(float(heightInFoot)*30.48
```

After dropping all the duplicates and dropping the columns where the missing values are very high, we start loading the data into the sql.

Analytical Database Design

In the Analytical part we have focused on EDA of mainly two points, player stats and team stats. We built tools to see player performance over the seasons and team performance by calculating stats like win percentage, points per game, points allowed per game and so on. By focusing on EDA, organizations can gain a deeper understanding of the data, uncover valuable insights, and make informed decisions to optimize player and team performance in the NBA.

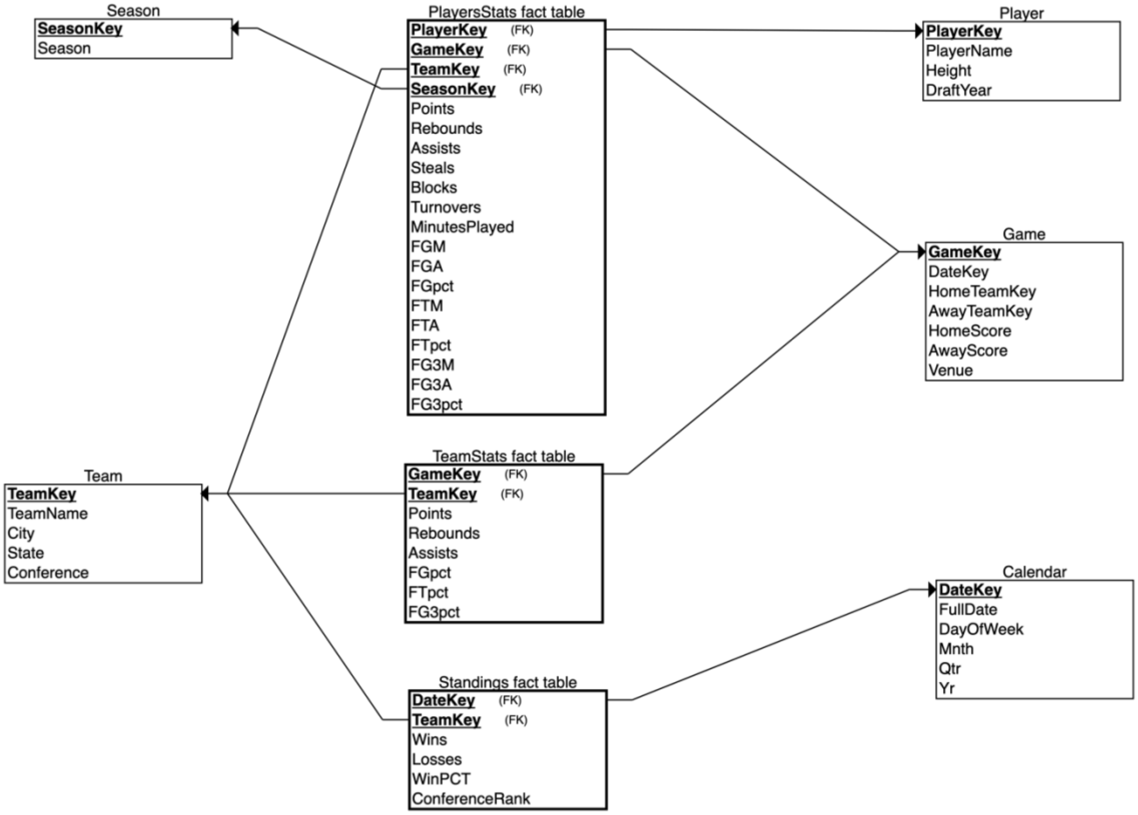


Fig 3. Star Schema of the Analytical Database

The user can analyze individual players, which players are most efficient, who improved in the game over time by seeing their scores in these visualization, and analyze by team , we can show the win percentage and scores per game defines the outcome of winning at home and away and with this user have the ability to slice and dice the stats by player and team and time reveals patterns, that how the coaches and general managers can use to maximize their odds of winning and we have taken this bunch of data turned it into an engine to improve strategy and performance.

ETL to Analytical Database

Loading calendar Table

```
sql = ( """
            INSERT INTO snps_wh.calendar(full_date, day_of_week ,
mnth, qtr, yr)
                SELECT DISTINCT(game_date), DAYNAME(game_date),
                       MONTH(game_date), QUARTER(game_date),
YEAR(game_date)
                FROM games;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading team Table

```
sql = ( """
            INSERT INTO snps_wh.team(team_key, team_name, city ,
state, conference)
                SELECT team_id, concat(city, " ", name), city,
                       state, conference
                FROM teams;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading player Table

```
sql = ( """
            INSERT INTO snps_wh.player(player_key, player_name,
height, weight, draft_year)
                SELECT player_id, player_name, height, weight,
draftyear
                FROM players;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading game Table

```
sql = ( """
            INSERT INTO snps_wh.game(game_key, date_key, home_team_key
, away_team_key, home_score, away_score, venue)
                SELECT G.game_id, C.date_key, home_team_id,
visitor_team_id, home.pts AS home_pts, away.pts AS away_pts, arena AS
venue
                FROM games G
                INNER JOIN game_stats home
                ON home.game_id = G.game_id AND home.team_id =
home_team_id
                INNER JOIN game_stats away
                ON away.game_id = G.game_id AND away.team_id =
visitor_team_id
                INNER JOIN teams T
                ON T.team_id = home_team_id
                INNER JOIN snps_wh.calendar C
                ON C.full_date =  G.game_date;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading player_stats Table

```
sql = ( """
            INSERT INTO snps_wh.player_stats(points, rebounds,
assists, steals, blocks, turnovers, minutes_played, fgm, fga, ftm,
fta, fg3m, fg3a, player_key, game_key, team_key)
                    SELECT pts, reb, ast, stl, blk, turnover,
seconds/60 as mintues_played, fgm, fga, ftm, fta, fg3m, fg3a,
player_id, TPGS.game_id, team_id
                    FROM snps_db.player_game_stats TPGS
                    INNER JOIN games G
                    ON G.game_id = TPGS.game_id;

        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading team_stats Table

```
sql = ( """
            INSERT INTO snps_wh.team_stats(points, rebounds, assists,
fg_pct,
                                                    ft_pct, fg3_pct,
game_key, team_key)
                SELECT pts, reb, ast, fg_pct, ft_pct, fg3_pct,
game_id, team_id
                FROM game_stats;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Loading standings Table

```
sql = ( """
            INSERT INTO snps_wh.standings(wins, losses, date_key,
                                                    team_key)
                SELECT GW, GL, date_key, team_id
                FROM snps_db.ranking
                INNER JOIN snps_wh.calendar
                ON full_date = standings_date;
        """
      )

cursor_operational.execute(sql)
conn_operational.commit()
```

Use Cases

This is the Welcome screen of the NBA Operational and Analysis database GUI application.

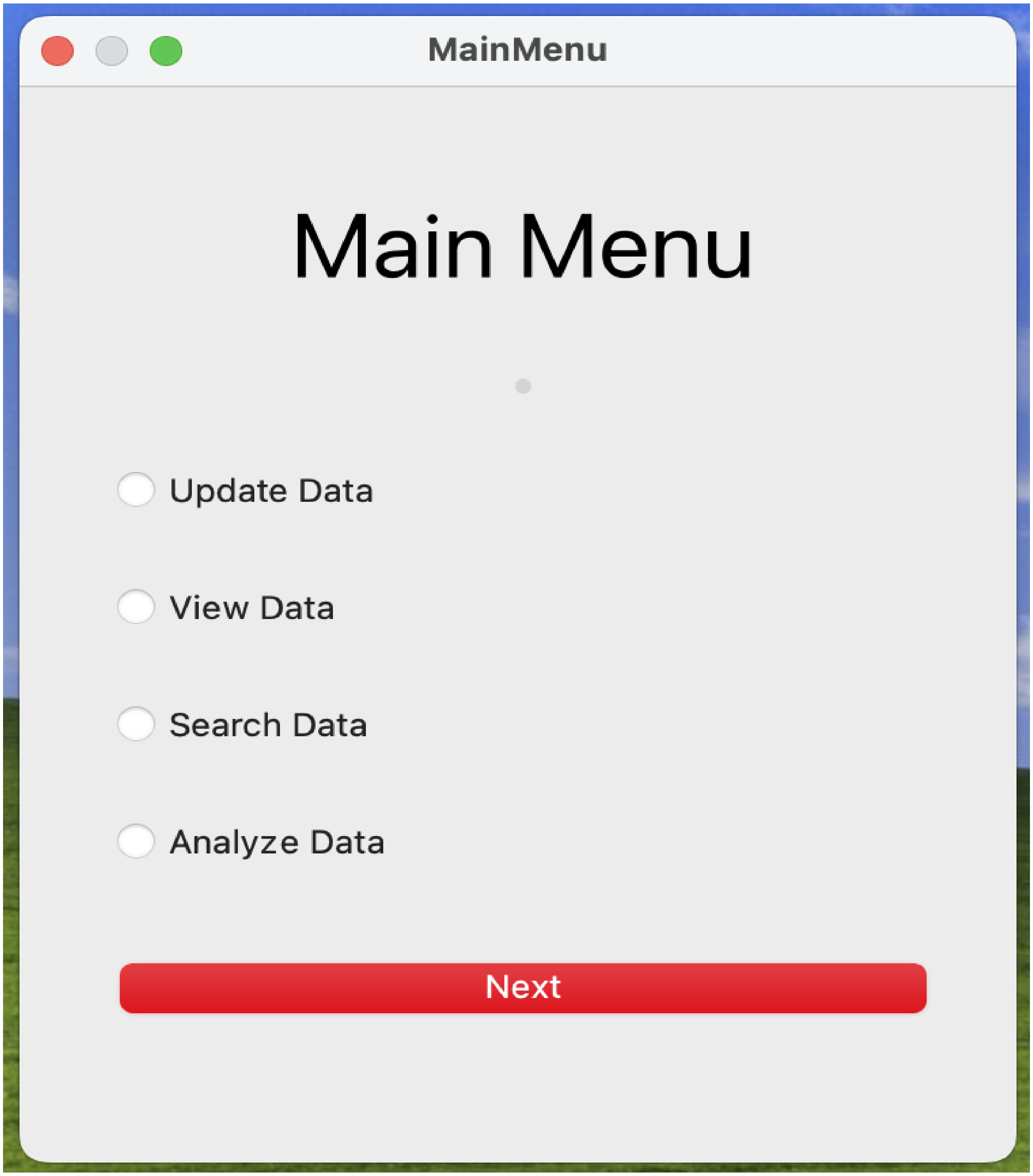


Fig 4. The above User Interface/Application gives the user a chance to Update Data, View Data, Search Data, and Analyze the Data.

User can use this window to update players table or team table.

Update Players

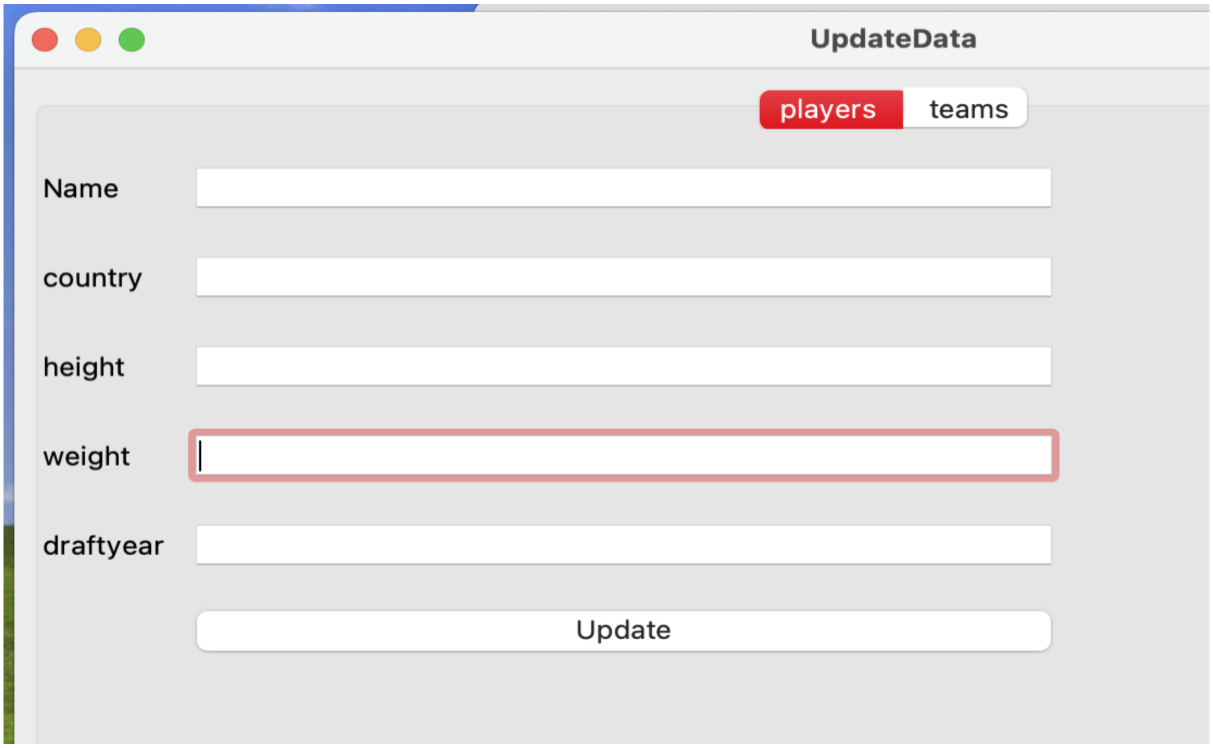


Fig 5. Adding new players who have entered the league.

Update Teams

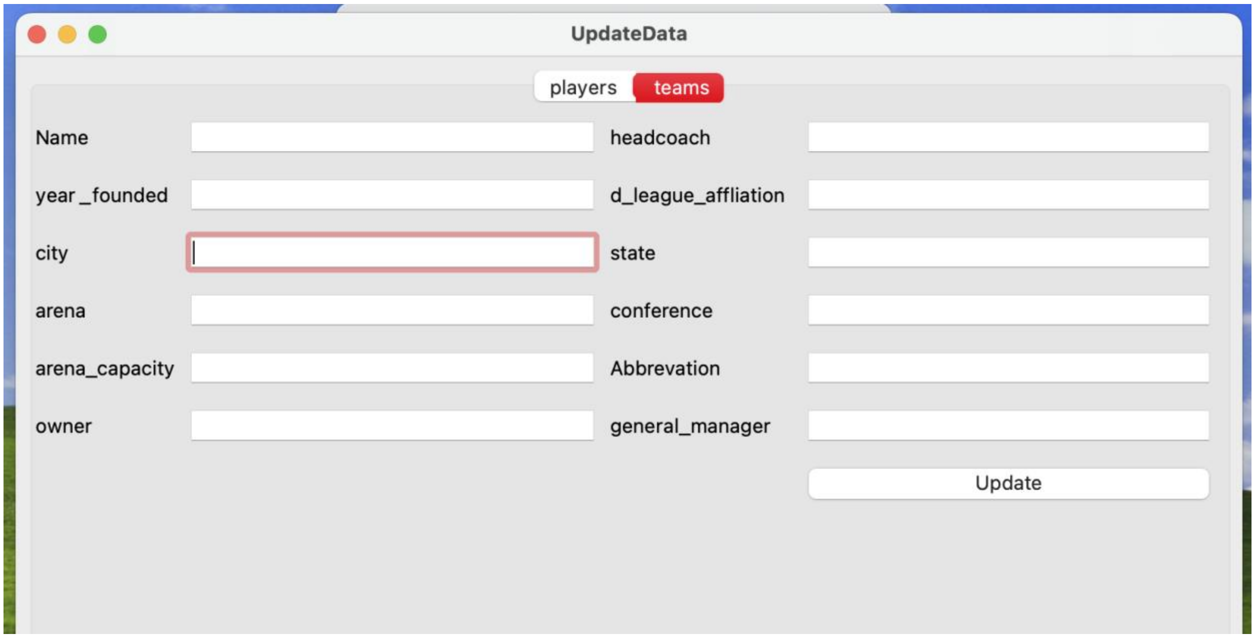
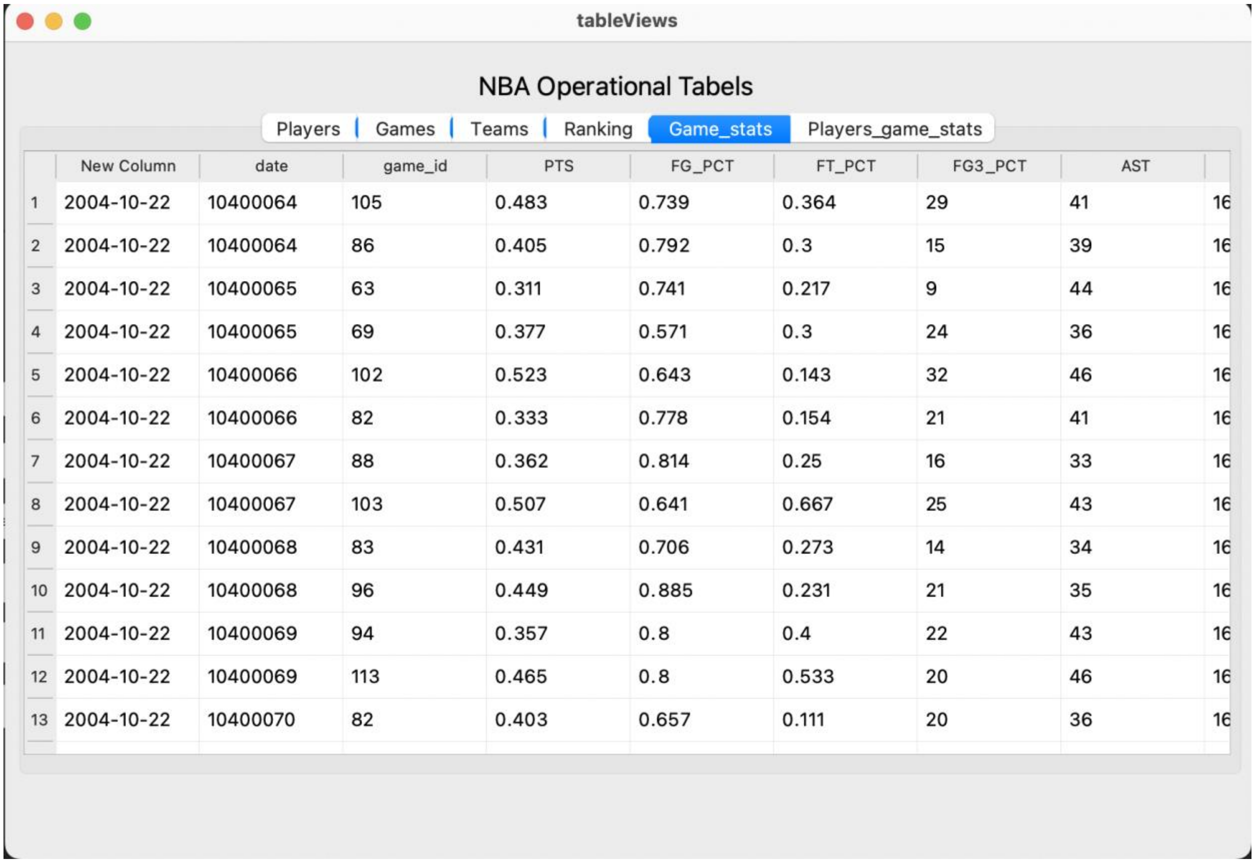


Fig 6. Adding new teams who have entered the league.

View Data



	New Column	date	game_id	PTS	FG_PCT	FT_PCT	FG3_PCT	AST	
1	2004-10-22	10400064	105	0.483	0.739	0.364	29	41	16
2	2004-10-22	10400064	86	0.405	0.792	0.3	15	39	16
3	2004-10-22	10400065	63	0.311	0.741	0.217	9	44	16
4	2004-10-22	10400065	69	0.377	0.571	0.3	24	36	16
5	2004-10-22	10400066	102	0.523	0.643	0.143	32	46	16
6	2004-10-22	10400066	82	0.333	0.778	0.154	21	41	16
7	2004-10-22	10400067	88	0.362	0.814	0.25	16	33	16
8	2004-10-22	10400067	103	0.507	0.641	0.667	25	43	16
9	2004-10-22	10400068	83	0.431	0.706	0.273	14	34	16
10	2004-10-22	10400068	96	0.449	0.885	0.231	21	35	16
11	2004-10-22	10400069	94	0.357	0.8	0.4	22	43	16
12	2004-10-22	10400069	113	0.465	0.8	0.533	20	46	16
13	2004-10-22	10400070	82	0.403	0.657	0.111	20	36	16

Fig 7. Window showing all the tables and the data.

Search Data

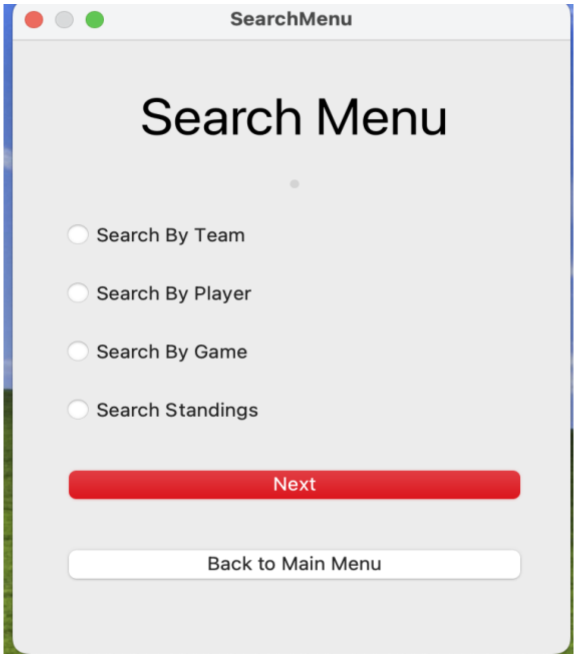
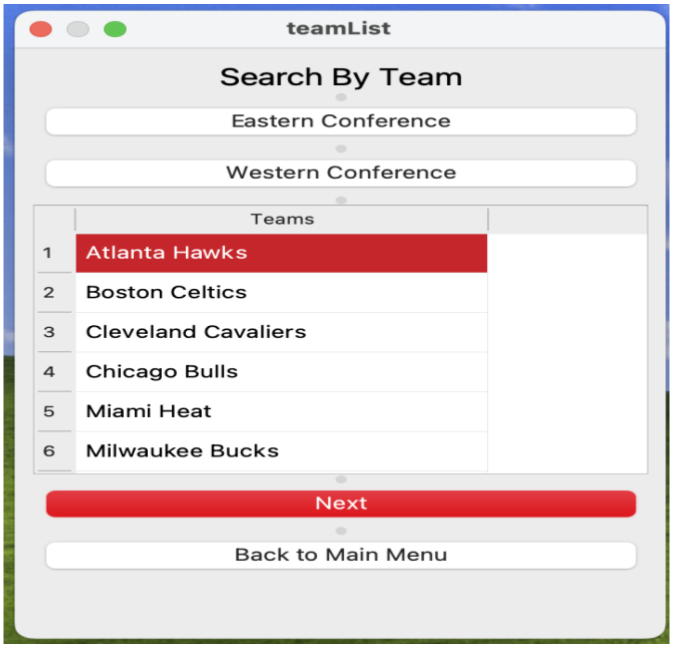


Fig 8. Menu to search using different parameters.

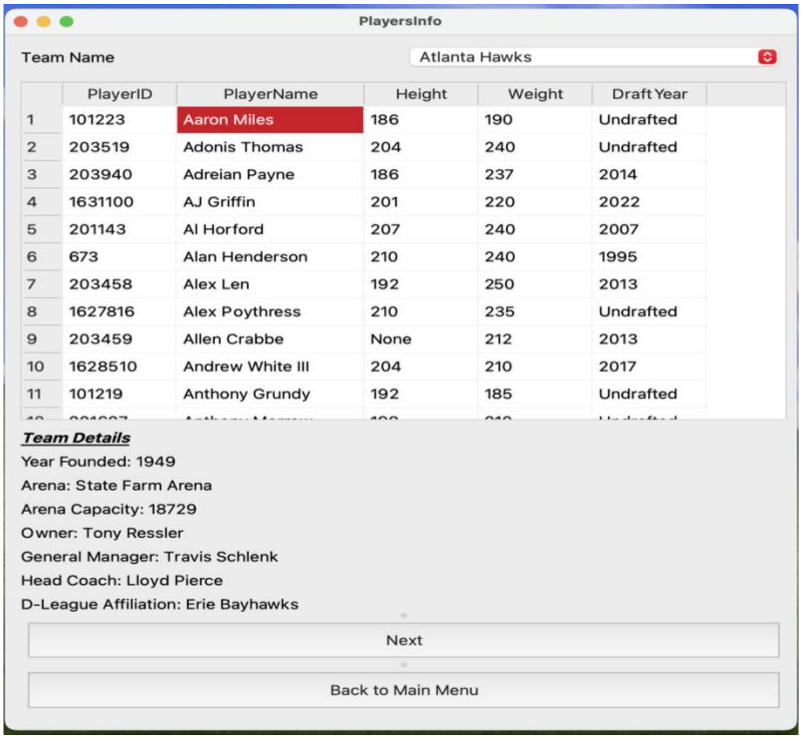
Search by Team

This window helps the user to find a player through the team the player has played for, on selecting the desired team name, list of players will be displayed who have played for that team and finally on selecting a player the player statistics of that player will be displayed.



	Teams
1	Atlanta Hawks
2	Boston Celtics
3	Cleveland Cavaliers
4	Chicago Bulls
5	Miami Heat
6	Milwaukee Bucks

Fig 9. List of teams in different conference



Team Name: Atlanta Hawks

	PlayerID	PlayerName	Height	Weight	Draft Year
1	101223	Aaron Miles	186	190	Undrafted
2	203519	Adonis Thomas	204	240	Undrafted
3	203940	Adreian Payne	186	237	2014
4	1631100	AJ Griffin	201	220	2022
5	201143	Al Horford	207	240	2007
6	673	Alan Henderson	210	240	1995
7	203458	Alex Len	192	250	2013
8	1627816	Alex Poythress	210	235	Undrafted
9	203459	Allen Crabbe	None	212	2013
10	1628510	Andrew White III	204	210	2017
11	101219	Anthony Grundy	192	185	Undrafted
12	201627	Anthony Morrow	198	210	Undrafted

Team Details

Year Founded: 1949

Arena: State Farm Arena

Arena Capacity: 18729

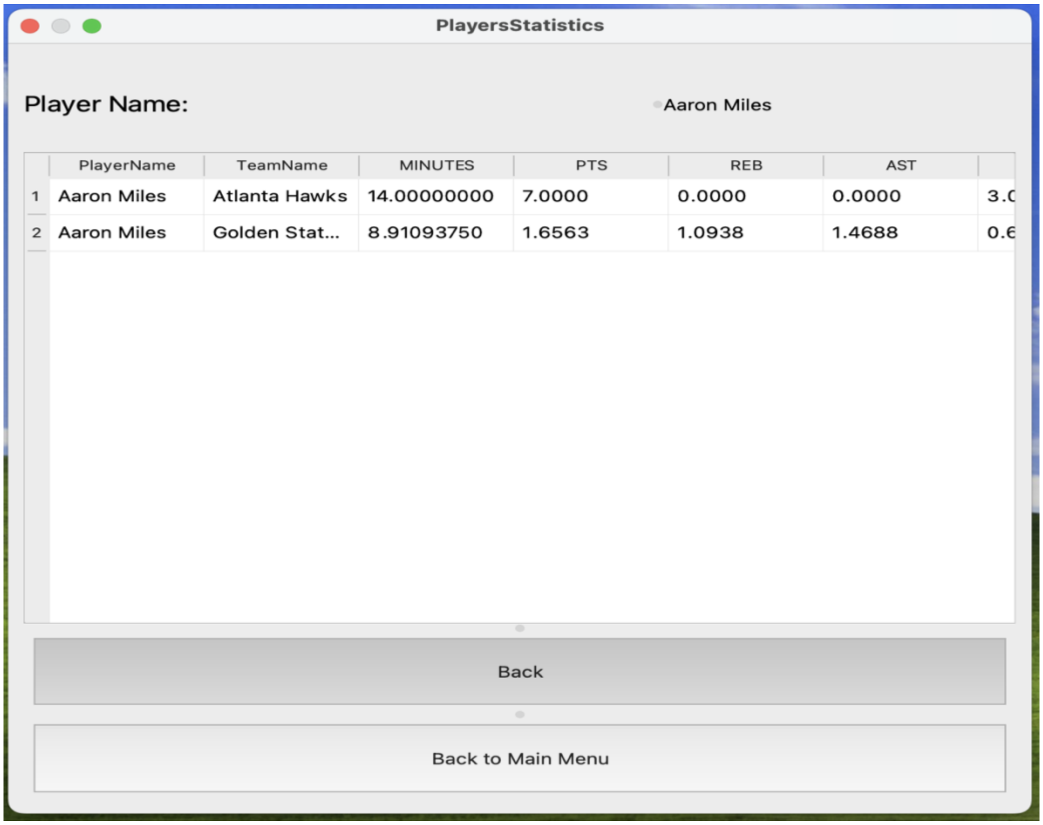
Owner: Tony Ressler

General Manager: Travis Schlenk

Head Coach: Lloyd Pierce

D-League Affiliation: Erie Bayhawks

Fig 10. All team members of the team selected in fig 9.



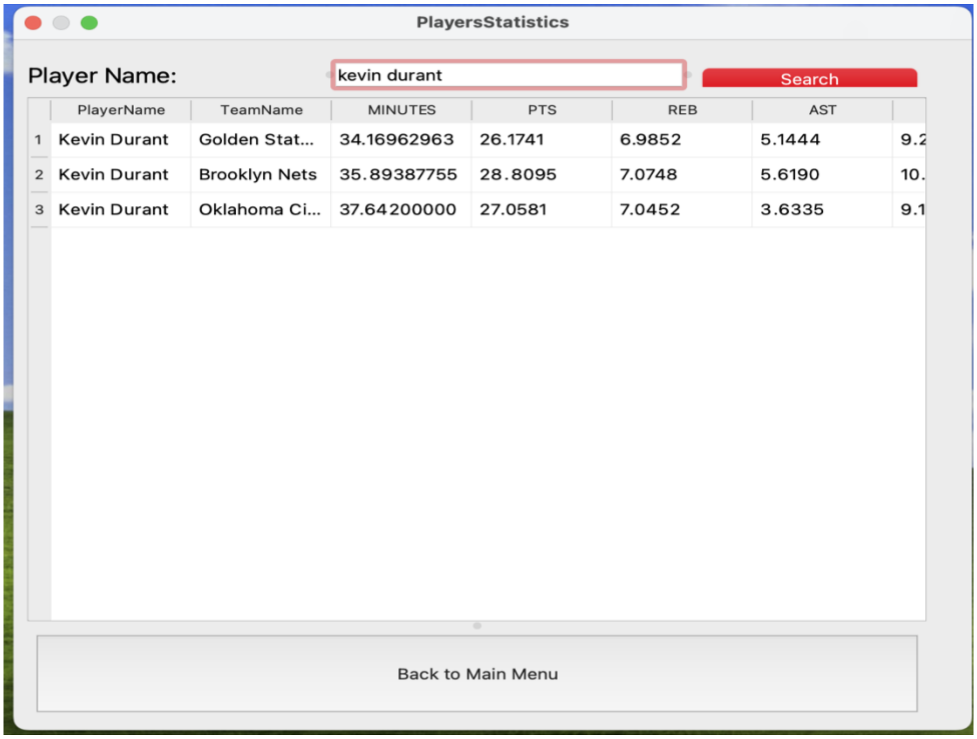
Player Name: Aaron Miles

	PlayerName	TeamName	MINUTES	PTS	REB	AST	
1	Aaron Miles	Atlanta Hawks	14.00000000	7.0000	0.0000	0.0000	3.0
2	Aaron Miles	Golden Stat...	8.91093750	1.6563	1.0938	1.4688	0.6

Fig 11. Player Statistics of the player selected in Fig 10.

Search by Player

Get player statistics over the season for different teams the player has played.

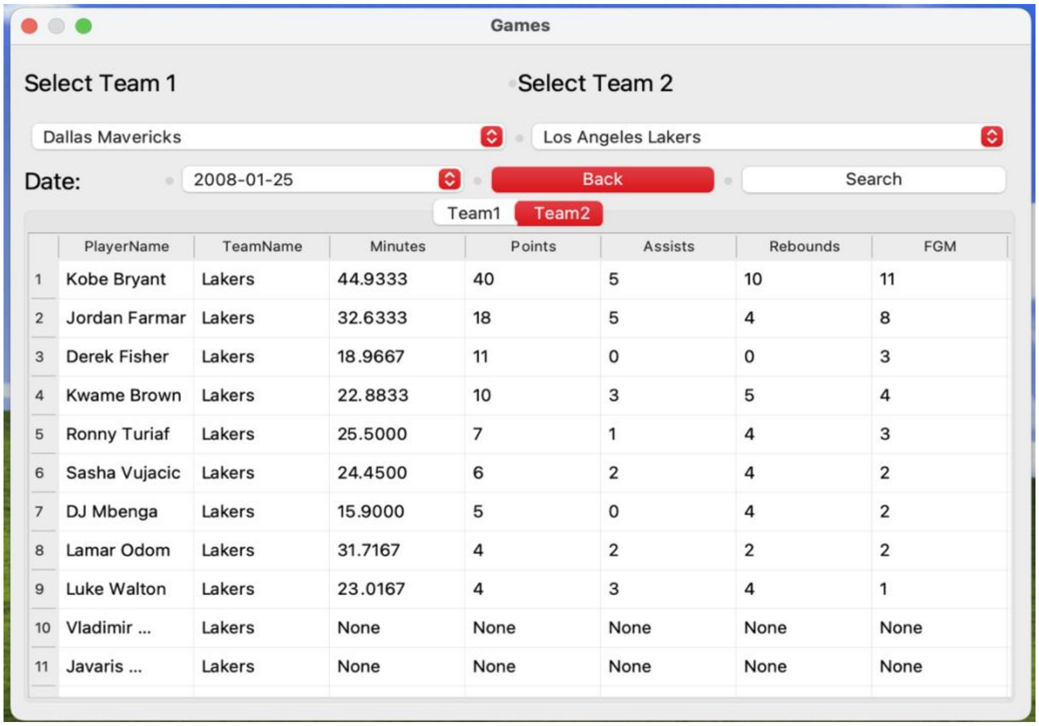


Player Name: kevin durant

	PlayerName	TeamName	MINUTES	PTS	REB	AST	
1	Kevin Durant	Golden Stat...	34.16962963	26.1741	6.9852	5.1444	9.2
2	Kevin Durant	Brooklyn Nets	35.89387755	28.8095	7.0748	5.6190	10.
3	Kevin Durant	Oklahoma Ci...	37.64200000	27.0581	7.0452	3.6335	9.1

Fig 12. Player Statistics of Kevin Durant

Search via Game



Select Team 1: Dallas Mavericks

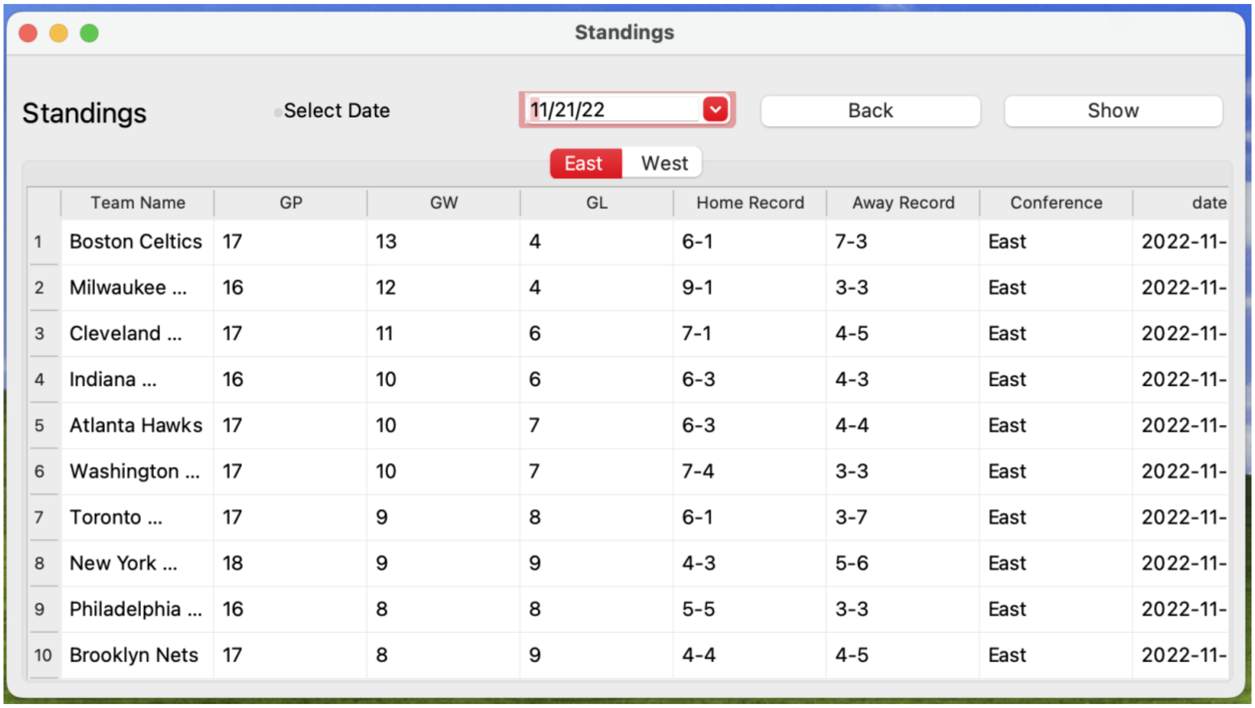
Select Team 2: Los Angeles Lakers

Date: 2008-01-25

	PlayerName	TeamName	Minutes	Points	Assists	Rebounds	FGM
1	Kobe Bryant	Lakers	44.9333	40	5	10	11
2	Jordan Farmar	Lakers	32.6333	18	5	4	8
3	Derek Fisher	Lakers	18.9667	11	0	0	3
4	Kwame Brown	Lakers	22.8833	10	3	5	4
5	Ronny Turiaf	Lakers	25.5000	7	1	4	3
6	Sasha Vujacic	Lakers	24.4500	6	2	4	2
7	DJ Mbenga	Lakers	15.9000	5	0	4	2
8	Lamar Odom	Lakers	31.7167	4	2	2	2
9	Luke Walton	Lakers	23.0167	4	3	4	1
10	Vladimir ...	Lakers	None	None	None	None	None
11	Javaris ...	Lakers	None	None	None	None	None

Fig 13. BoxScore of a game played by Team1 and Team2 seleted by the user on Date.

Search by Standings



Standings

Select Date 11/21/22

	Team Name	GP	GW	GL	Home Record	Away Record	Conference	date
1	Boston Celtics	17	13	4	6-1	7-3	East	2022-11-
2	Milwaukee ...	16	12	4	9-1	3-3	East	2022-11-
3	Cleveland ...	17	11	6	7-1	4-5	East	2022-11-
4	Indiana ...	16	10	6	6-3	4-3	East	2022-11-
5	Atlanta Hawks	17	10	7	6-3	4-4	East	2022-11-
6	Washington ...	17	10	7	7-4	3-3	East	2022-11-
7	Toronto ...	17	9	8	6-1	3-7	East	2022-11-
8	New York ...	18	9	9	4-3	5-6	East	2022-11-
9	Philadelphia ...	16	8	8	5-5	3-3	East	2022-11-
10	Brooklyn Nets	17	8	9	4-4	4-5	East	2022-11-

Fig 14. Standings of the league on a date selected by the user.

Analyze Data

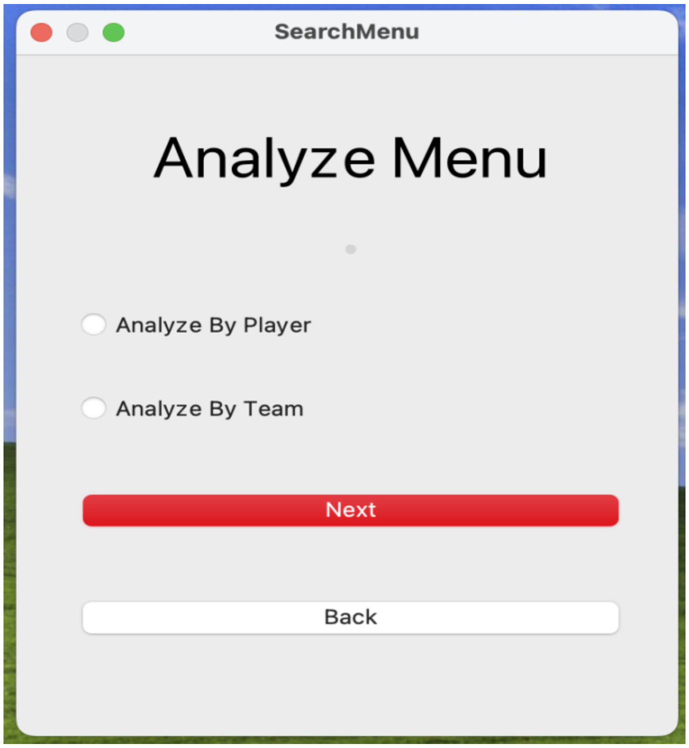
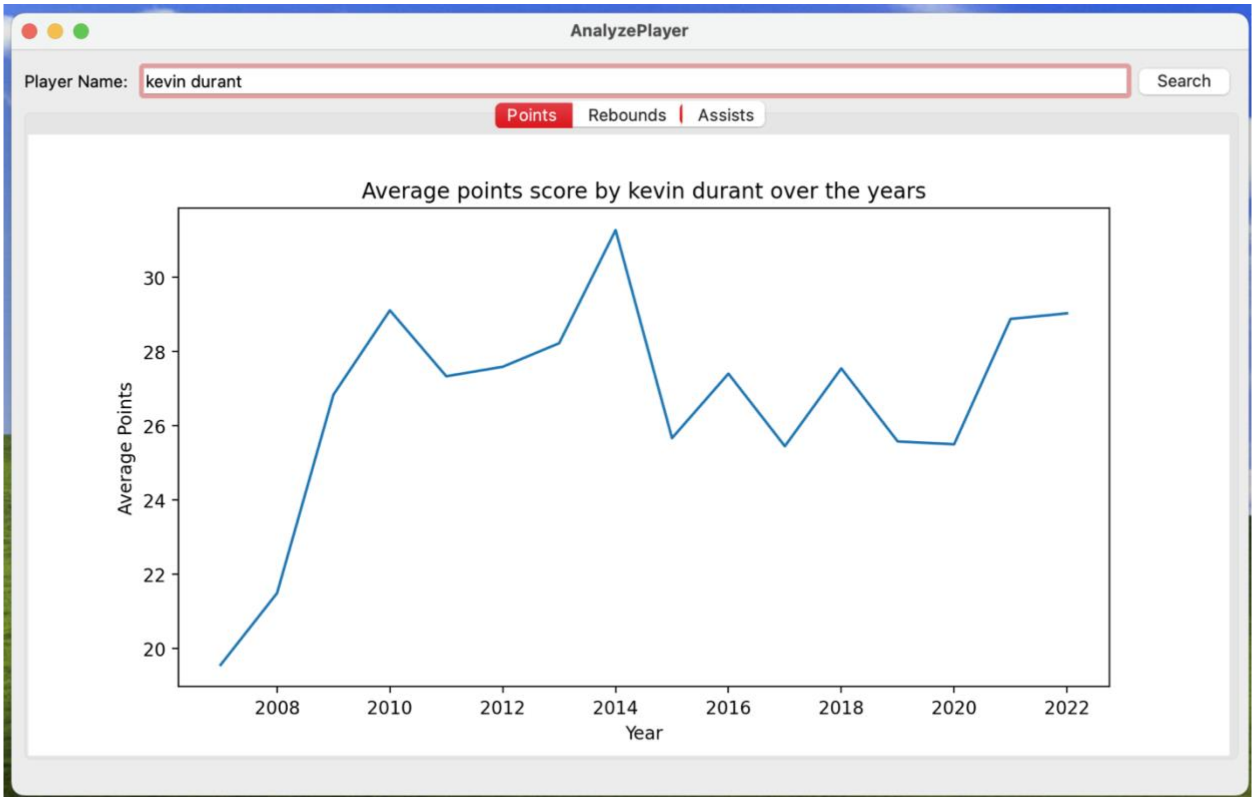


Fig 15. Menu for Analyzing data either by Player or Team

Analyze by Player

This shows the player performance over the years, performance metrics include Points, Rebounds and Assists and the aggregation done here is Average.

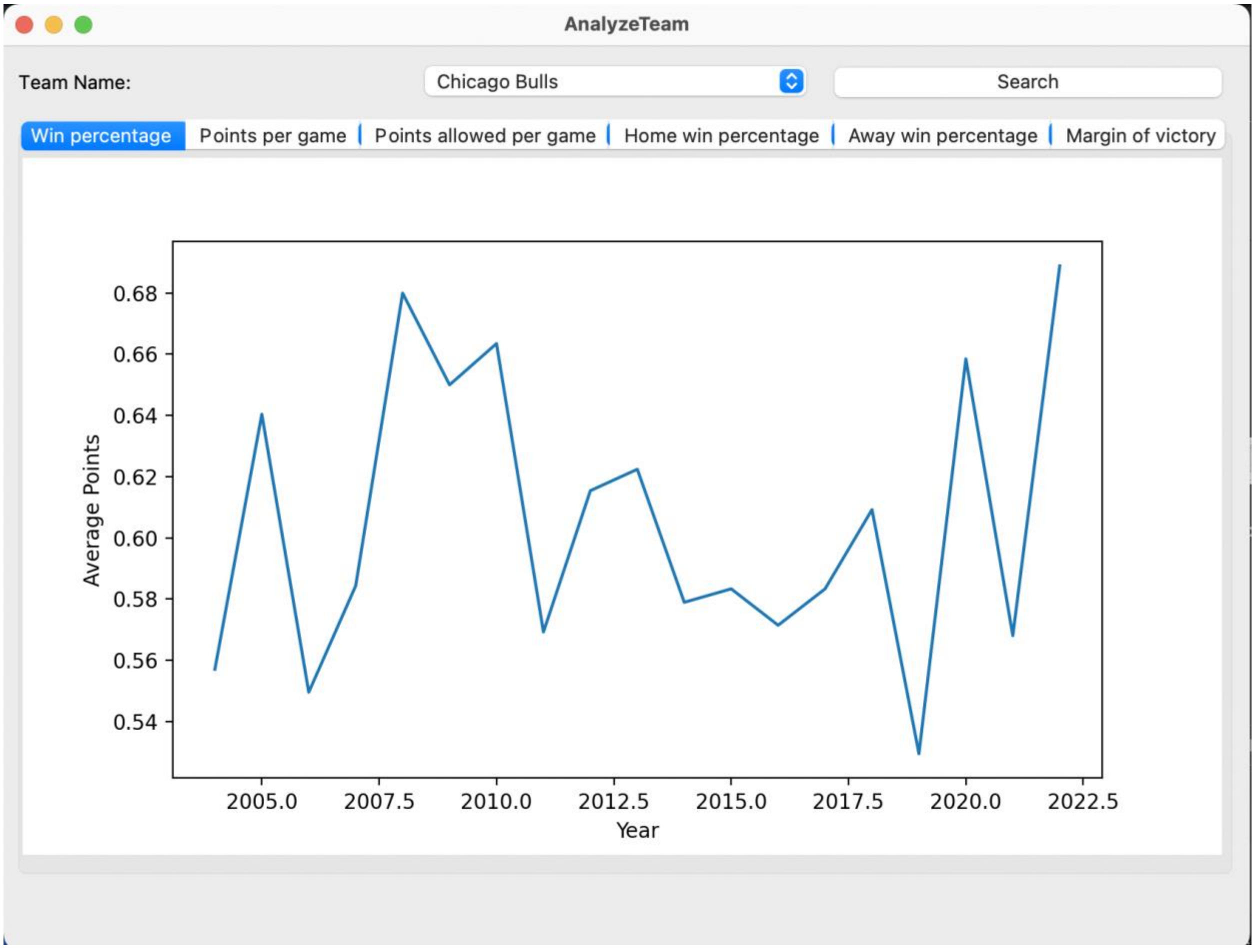


Player Name: kevin durant

Fig 16. Line graph of Average Points per year for Kevin Durant

Analysis by team

This shows the team performance over the years, performance metrics include Win percentage, Points per game, Points allowed per game, Home win percentage, Away win percentage and Margin of Victory.



Team Name: Chicago Bulls

Fig 17. Line graph of the Win Percentage of Chicago Bulls over the years.

References

[1] https://www.kaggle.com/datasets/nathanlauga/nba-games

[2] https://www.nba.com/stats

[3] https://github.com/swar/nba_api