CSC 336 Database Systems

Runtime Terror
Software Documentation
For NBA Reference

Version 1.0

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this software is to provide an all-in one source to review NBA player game statistics as well as give the user the option to choose their favorite players and teams. The software will include up-to-date statistics. It will also provide an interactive GUI that the user can scroll and tap through, which creates an immersive and easy-to-follow experience. NBA fans always wanted to have more knowledge about the most up-to-date statistics without the hassle of needing to search their favorite teams or players individually every time. With this system, users can check back at any point to review player stats, which are updated to match records up to the current season, observe team standings, and pick their favorites which would appear on their main screen.

1.2 Scope

Whether you've been a fan for years or saw your first basketball game this morning, this app is friendly for everyone interested in the NBA. Many people who are unfamiliar with the NBA will learn about all the teams that exist, along with their respective players. This software makes sure that the user will learn about what each statistic means and how they determine skill level. Fans of the NBA will realize that this software is perfect for viewing up-to-date information on their favorite players and teams.

1.3 Definitions, Acronyms, and Abbreviations

City	City the NBA Team is located in	**RPG**	The amount of rebounds a player averages per game
Division	Geographic location in the U.S. for NBA Team	**APG**	The amount of assists a player averages per game
Conference	NBA Team belongs to either East or West Conference	**SPG**	The amount of steals a player averages per game
GP	Total games played by a player	**BPG**	The amount of blocks a player averages per game
PPG	The amount of points a player averages per game	**TOV**	The amount of turnovers a player averages per game

2. Database Documentation

2.1 SQL Script

```
DROP DATABASE IF EXISTS 336_project;
CREATE DATABASE 336_project;
USE 336_project;

DROP TABLE IF EXISTS Team;
CREATE TABLE Team(
       Name VARCHAR(255) PRIMARY KEY,
       City VARCHAR(255),
       Division VARCHAR(255) NOT NULL,
       Conference VARCHAR(255) NOT NULL,
       Wins INT NOT NULL,
       Losses INT NOT NULL
);

DROP TABLE IF EXISTS Players;
CREATE TABLE Players(
       Id INT PRIMARY KEY,
       Name VARCHAR(255) NOT NULL,
       Birthdate DATETIME NOT NULL,
       Height VARCHAR(255) NOT NULL,
       Position VARCHAR(255) NOT NULL,
    Team VARCHAR(255),
       FOREIGN KEY(Team) REFERENCES Team(Name)
);

DROP TABLE IF EXISTS PlayerStats;
CREATE TABLE PlayerStats(
       GP INT,
       PPG DECIMAL(3,1),
       RPG DECIMAL(3,1),
       APG DECIMAL(3,1),
    SPG DECIMAL(3,1),
       BPG DECIMAL(3,1),
    TOV DECIMAL(3,1),
    PlayerID INT PRIMARY KEY,
       FOREIGN KEY(PlayerID) REFERENCES Players(Id)
);

DROP TABLE IF EXISTS User;
CREATE TABLE User(
       Id INT PRIMARY KEY,
       Name VARCHAR(255) NOT NULL,
       Email VARCHAR(255) UNIQUE NOT NULL,
```

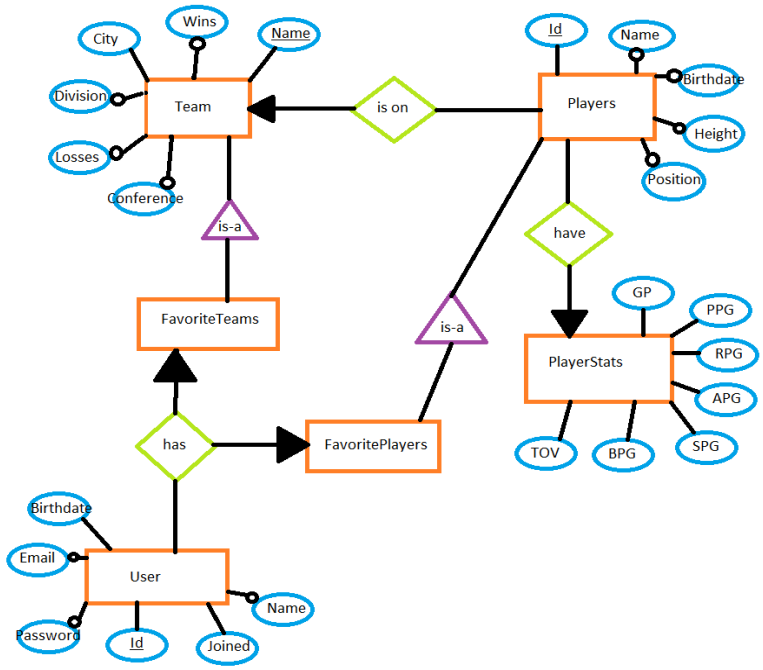
```
      Birthdate DATETIME,
      Password VARCHAR(255) NOT NULL,
    Joined DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    IsSignedIn BOOLEAN NOT NULL
);

DROP TABLE IF EXISTS FavoritePlayers;
CREATE TABLE FavoritePlayers(
     playerID INT,
     userID INT,
     FOREIGN KEY(playerID) REFERENCES Players(ID),
     FOREIGN KEY(userID) REFERENCES User(ID),
     PRIMARY KEY(userID, playerID)
);

DROP TABLE IF EXISTS FavoriteTeams;
CREATE TABLE FavoriteTeams(
     teamName VARCHAR(255),
     userID INT,
     FOREIGN KEY(teamName) REFERENCES Team(Name),
     FOREIGN KEY(userID) REFERENCES User(ID),
     PRIMARY KEY(userID, teamName)
);
```

2.2 Schemas

Details for each schema can be viewed in the SQL script above. The schemas used were:

- Team
- Players
- PlayerStats
- User
- FavoritePlayers
- FavoriteTeams

2.3 ER Diagrams



3. Software Documentation

This software utilizes Python 3.8, MySQL, and additional Python packages, which can be installed in the terminal. Below are the required packages:

1) Access to NBA API

```
pip install nba_api
```

2) Access to Pandas

```
pip install pandas
```

3) Access to Pandas Table

```
pip install pandastable
```

4) Access to MySQL Connector for Python

```
pip install mysql-connector-python
```

3.1 How to Start Application

Make sure all the necessary files are downloaded and included within the same folder in the project directory. These files are:

- nba_data.py
- nba_sql_commads.py
- nba_gui.py
- **main.py**
- nba_database.sql
- logos (folder that contains team logos)

The program will be run using the main.py file. This software requires the use of a database connected to MySQL. The username and password are set to default values in the main.py file, but please change if necessary. Running the main.py file will bring up the GUI, which has its own guide in Section 3.2.

3.2 Windows and Menus Guide

Upon starting the application, the user will be met with the main menu window. On the top, there are many dropdown menus to select from. On the right, there is a refresh button that updates the main window.

There are 5 demo users already, with their own favorite players and teams. The credentials for each are below:

Ash) Email: ash@gmail.com, Password: Ash

JC) Email: jc@gmail.com, Password: JC

Eric) Email: eric@gmail.com, Password: Eric

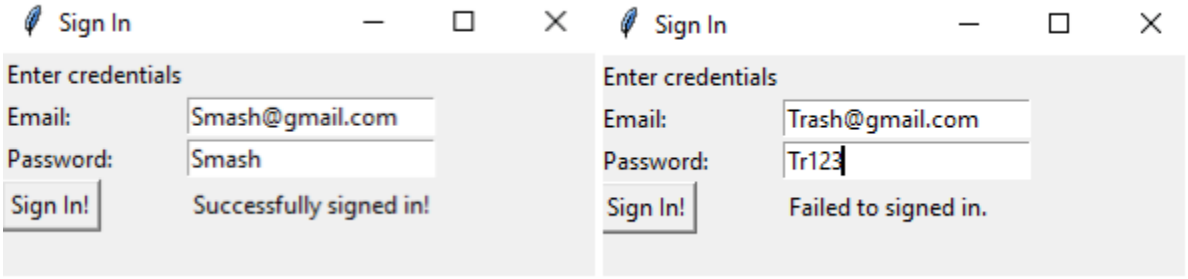
Blue) Email: blue@gmail.com, Password: Red

Red) Email: red@gmail.com, Password: Blue

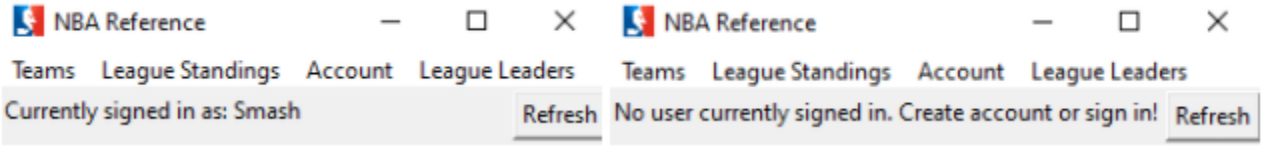
3.2.1 Account

Account is where the user can choose to sign in, sign out, view, or delete their account.

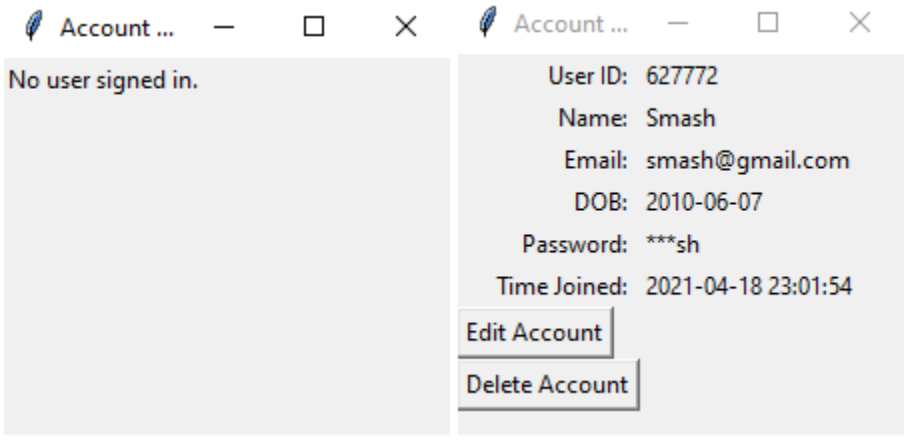
1) “Sign In” brings the user to a menu where they can input their email and password to sign in. If their credentials are correct, they’ve successfully signed in and the main window is updated to show this user. If not, they’re given a message indicating that they’ve failed to sign in.



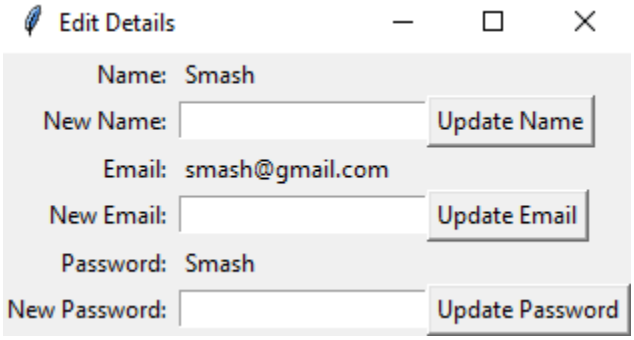
2) “Sign Out” simply removes the user from the database and updates the main window so that it displays no users are signed in. Below is a before and after of pressing “Sign Out”.



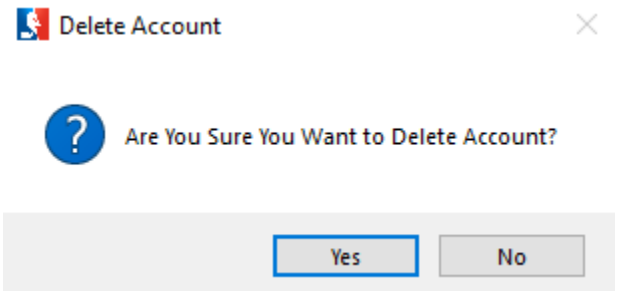
3) “Account Details” brings up a new window in which the user can check their signed in user profile details. If there are no users signed in, it will be displayed as a label.



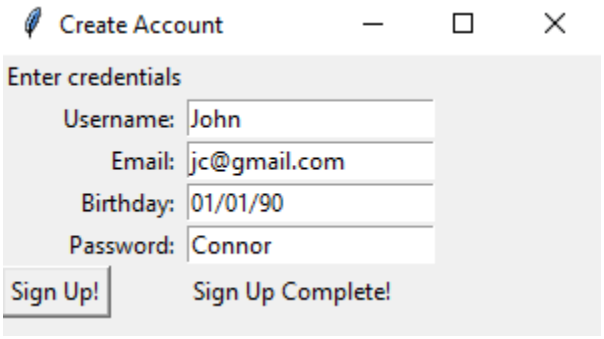
As seen above, there are two additional buttons for editing account and deleting account. Pressing “Edit Account” will bring the user to a new window where they can view their essential details and change whichever detail they’d like.



Going back to the previous window, pressing “Delete Account” will give the user a prompt to select either “Yes” or “No” if they are sure they want to delete their account.

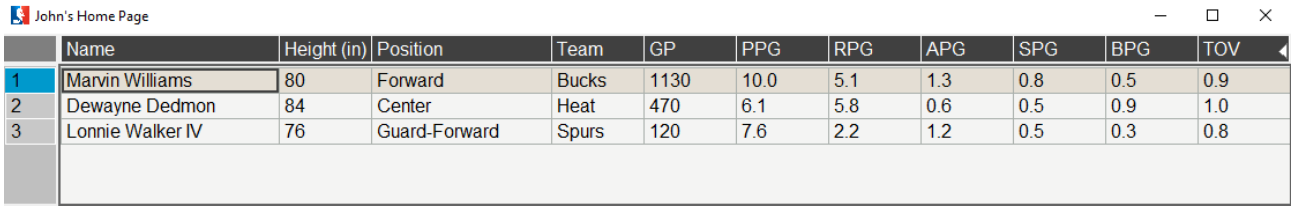


4) “Create Account” brings up a new window in which the user can create a new account by putting in their credentials. A duplicate email and password combo will display a warning, but a unique combo will result in a successful sign up. The main window will also be updated. Birthdays need to be inputted in MM/DD/YY format.



3.2.2 Home

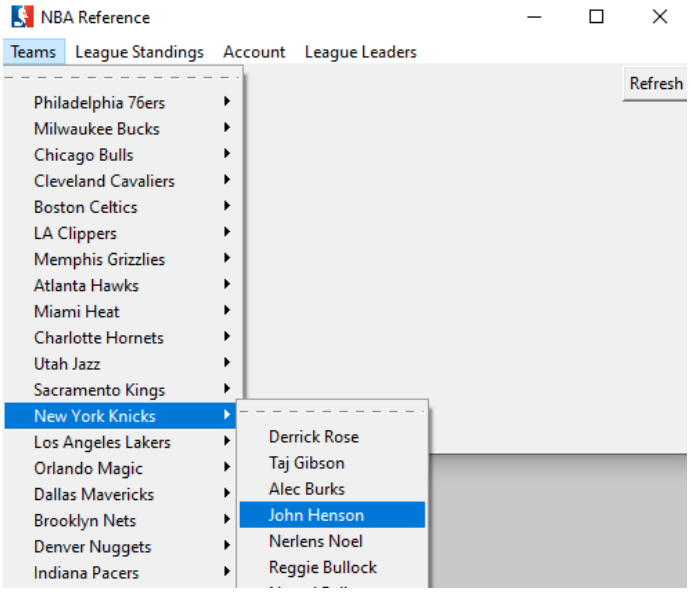
Home brings up a window that displays the signed in user's favorite teams and players. An example is shown below for a user named "John".



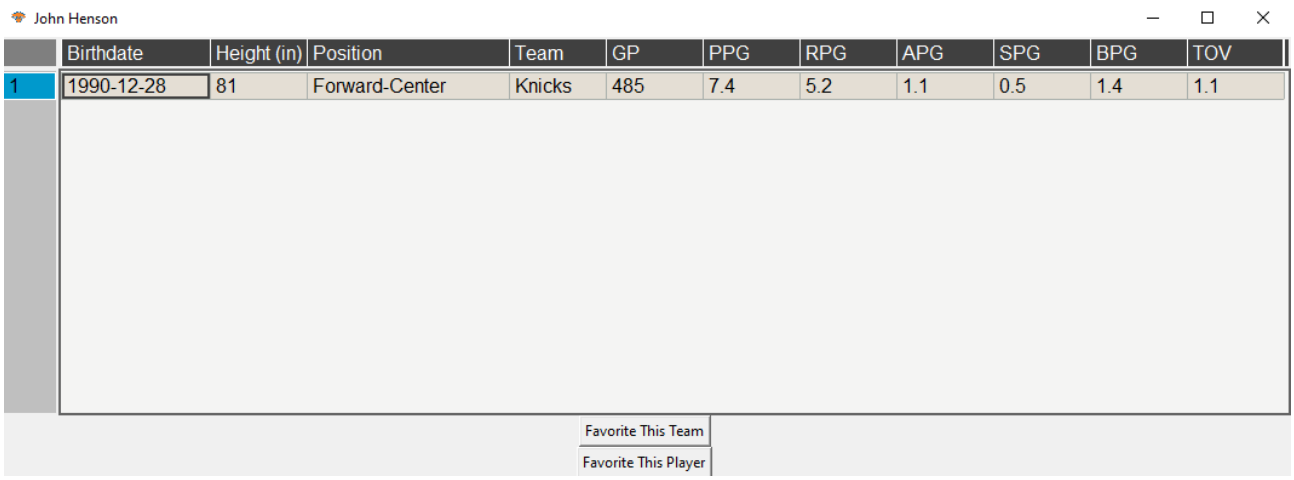
	Name	Height (in)	Position	Team	GP	PPG	RPG	APG	SPG	BPG	TOV
1	Marvin Williams	80	Forward	Bucks	1130	10.0	5.1	1.3	0.8	0.5	0.9
2	Dewayne Dedmon	84	Center	Heat	470	6.1	5.8	0.6	0.5	0.9	1.0
3	Lonnie Walker IV	76	Guard-Forward	Spurs	120	7.6	2.2	1.2	0.5	0.3	0.8

3.2.3 Teams

Teams is where the user can view every team in the NBA, along with their respective players.



Clicking on a particular player in a team brings up a new window that displays their overall stats.



On the bottom, there are two buttons: one for favoriting the team and the other for favoriting the player. Pressing either button will change the state of the text so that the user has the option to unfavorite whatever they had chosen to favorite before. This works vice versa, so the user can always toggle between favoriting a team or player, or unfavoriting.

3.2.4 League Standings

League Standings displays how well each team performed in the league.

1) “League” will bring up a new window that displays every single team in the league alongside their information (location, name, wins, losses, etc.), no matter what conference they belong to. The teams are sorted by number of wins.

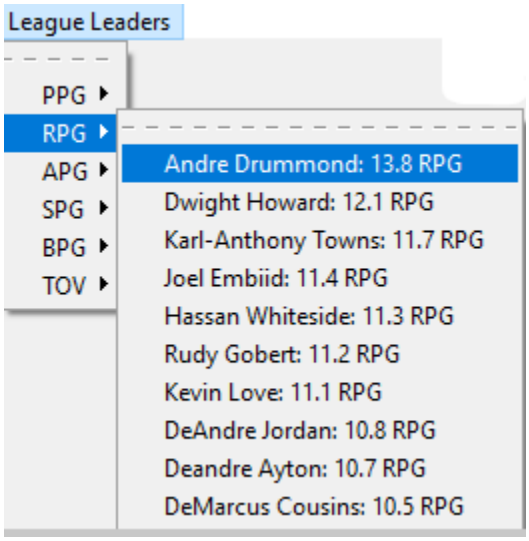
	Name	City	Division	Conferenc	Wins	Losses
1	Jazz	Utah	Northwest	West	40	13
2	Suns	Phoenix	Pacific	West	37	15
3	Clippers	LA	Pacific	West	37	18
4	76ers	Philadelphia	Atlantic	East	36	17
5	Nets	Brooklyn	Atlantic	East	36	17
6	Nuggets	Denver	Northwest	West	34	19
7	Bucks	Milwaukee	Central	East	33	20
8	Lakers	Los Angeles	Pacific	West	33	20
9	Trail Blazers	Portland	Northwest	West	31	22
10	Mavericks	Dallas	Southwest	West	29	23
11	Hawks	Atlanta	Southeast	East	29	25
12	Heat	Miami	Southeast	East	28	25
13	Celtics	Boston	Atlantic	East	28	26

2) “East” will display teams who are only in the Eastern Conference. The teams are sorted by number of wins as well.

3) “West” will display teams who are only in the Western Conference. The teams are sorted by number of wins as well.

3.2.5 League Leaders

League Leaders displays the top players in each category for the league.



Each category is listed as PPG, RPG, APG, SPG, BPG, and TOV, each representing a stat for a player. Hovering over each stat will display the top 10 players who performed well for that category by having the highest number.

3.3 Supplementary Requirements

3.3.1 Favorites Feature

Other sports applications allow the user to only view their NBA teams of choice, along with some of their players. Not only does this program offer up-to-date data, but it also features a favoriting system. Users can click on individual players and decide if they want to favor this particular player and/or the team. They have the choice of choosing what teams and players they like, and they can view their favorites using the Home tab.

3.3.2 Logos

To give the application a fresh look, the user can view the NBA logo on start up on the top left corner of their window. Clicking on a player in a team will also display the team logo on the top left, which gives every window a fresh look.

3.3.3 Choice of Stats Display

Normally in this GUI, a way to display statistics or data would be to use grids in the window. But an alternative and cleaner method was considered, and later implemented in most of the windows. The pandastable import gave the option to display data in an easy to read format. It is useful for updating data to the tables and the scrolling feature makes the user experience better.

4. Closing Notes

4.1 References

Python GUI with Tkinter Playlist - https://youtube.com/playlist?list=PL6gx4Cwl9DGBwibXFtPtflztSNPGuIB_d

Installing NBA API - https://pypi.org/project/nba-api/

Github for Referencing NBA API - https://github.com/swar/nba_api

4.2 Challenges Encountered and Solutions

4.2.1 Python

Python was a language the team was barely knowledgeable of. We were well versed in the Java and C field, but we decided that going with Python would not only be a good learning experience, but it would also prove to be suitable for pulling data from the API that we wanted, as well as creating a GUI using Tkinter. It was manageable to deal with both frontend and backend details using Python, even though it proved to be challenging to learn new concepts and visiting the Tkinter library for the first time.

4.2.2 API

APIs are filled with an incredible amount of information for developers around the world to use in their own projects. We were still new to Python, and even more to using APIs using Python. Thankfully, the main NBA API website had a guide on how to install their API on our Python editing platform, but we had trouble picking out what specific data we wanted.

There were dozens of dataframes to choose from, and hundreds of individual variables and keys needed to access some of these frames. Searching through the documentation proved to be tasking, since we weren't sure on what statistics would be the most relevant for users who used the app. In the end, we went with tracking player and team data, and pulled all the information by fetching from the NBA API database and adding it to our own. It took time to process all the information and cherry pick out specific traits that we wanted, especially since the API doesn't allow many requests all at once, but the data proved to be useful, and meets the requirements that we desired.

4.2.3 Time Management

Addressing the issue of learning Python and pulling from the API took a good chunk of the team's time. We could've released this application by the first deadline, but we didn't want to send out a half baked program without hitting most of the goals we had in mind. That's why we requested for the extension; we were determined to make this program appear and behave more user friendly. With this extended period of time, we were able to finish the Favorite Teams/Players portion of our code, as well as tweak some other parts like Account Details, League Standings, and window proportions. We appreciate the extra time given to us and we're thankful for the opportunity to improve.

4.2.4 User Persistence and Signing In/Out

Having users be a part of a program increases the chance of them wanting to use it for

longer. Users would have the choice to choose who they favorite and view them whenever they please. An issue that we encountered is persistence, which means keeping track of who was last using the app and what data they've changed. We encountered the issue of logging in and picking a player to add to their favorites tab. But when we closed the program and re-opened it, the favorite player would be gone from their account, and the user would be signed out. To fix this, we needed to add a column for checking who is signed in in our User table. Anyone who is signed out would have a value of 0, and the one user who is signed in would have a value of 1.

4.3 Future Plans

4.3.1 Inclusion of Former NBA Players

In the future we would like to include former NBA players. This would allow users to find the stats of their favorite players from the past!

4.3.2 Inclusion of More Stats

Another addition we would love to make is the inclusion of more stats. From basic stats such as +/- to more advanced metrics like PER (player efficiency rating). This makes the application useful to the common fan and to the more analytical fans.

4.3.3 WNBA Expansion

For the diehard basketball fans who love the sport as a whole, we would include the teams, players, and stats of players that are in the WNBA. This allows for WNBA fans to have access to both womens and mens professional basketball statistics all in the same place.

4.3.4 Player Accolades

This addition would make it possible for the user to see what accolades a player has received at any point of their basketball career. This can also allow us to create lists of previous award winners, runner ups, and even predictions for who the next award winner will be.

Update favorite players for updated players (players leaving, retired, etc.)

4.3.5 Olympic Teams

Because the United States Olympic Basketball Team is composed of mostly, if not entirely, NBA Players, showing the current and previous Olympic Teams would allow for users to see if their favorite players had ever played in the Olympics.

4.3.6 Live Updating/Refreshing

With the ability to update live, the stats of players would change on a daily basis, making the application as precise as the last game that was played. This gives the best user experience as stats are consistently up-to-date.

4.3.7 API Usage

Figuring out a way to pull data from the NBA API without waiting so long would make testing, updating, and editing data much more convenient and easy. The NBA API doesn't allow multiple requests at once, so a sleep timer was necessary to put in between requests. Without it, an error

would occur. This sleep timer is the reason that running the program for the very first time takes a while. We want users to be able to access the data they desire instantaneously. Finding a way to remove or decrease the wait time would increase user satisfaction.

4.3.8 Additional Online Features (Multiple Users and Discussion Boards)

Making this application available to multiple users at a time would make it more engaging. Having a discussion board under each player or team would allow users to interact with each other and give input on why they chose them as their favorite (or why they did not).