**Group Project Proposal Revision   
CIS 560: Database System Concepts**   
Jacob Koelzer Kaden O’Hare Isaac Peterson

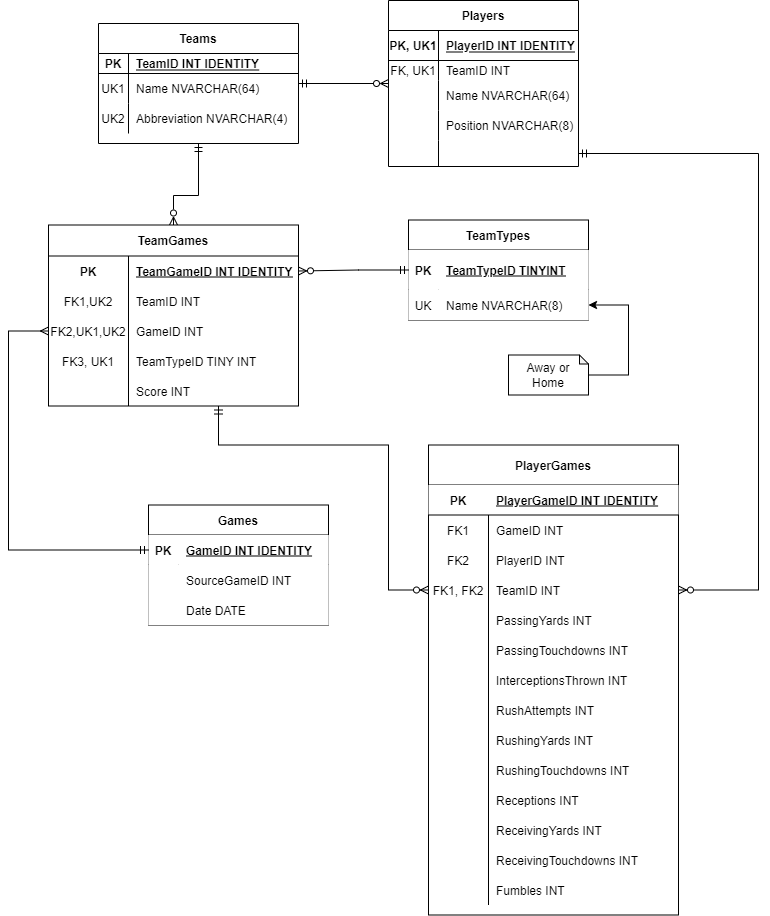
**Introduction:**

This program is a tool for football fans or analysts to view a variety of stats pertaining to a season which includes game, team, and player stats. The user can choose between these types of stats in the GUI. There is a search option to find a specific player or team. A game can be found easier by filtering on the date.

Clicking on a player displays the stats for that player over the entire season. The total stats for a team can be viewed in a similar fashion. There’s also a button to view the top players over the entire season in terms of touchdowns, which can be filtered on total, passing, rushing, and receiving.

**Technical Description:**

The user interface is a windows forms desktop app with its logic layer written in C#. The backend database was made in Microsoft SQL Server Management Studio. The library System.Data.sqlclient was used to connect to the database.

**Database Design:**  
   
****

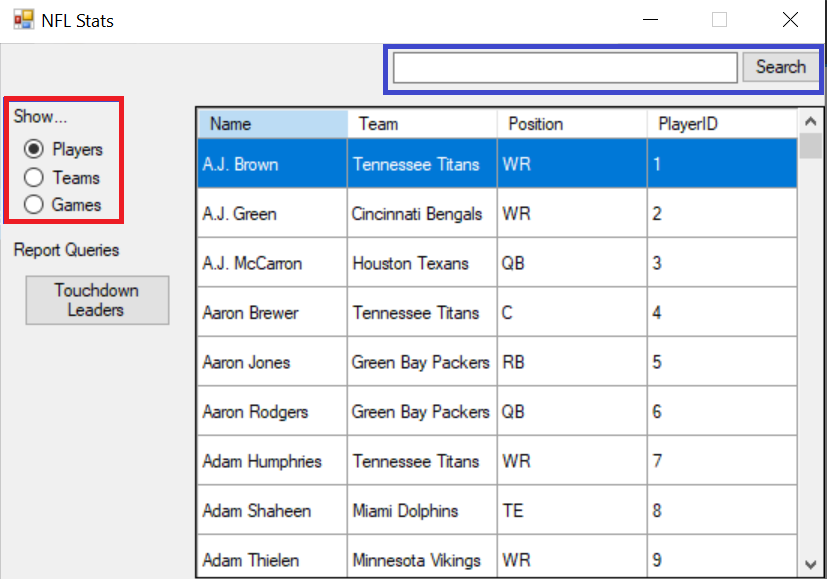
**System Design:**

Graphical user interface, text, application

Description automatically generated

**System Features and Usage:**

On launch the user should get a screen similar to this:



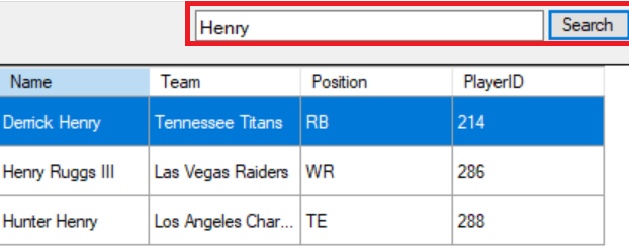
The buttons circled in red allow the user to view a list of players, teams, or games. The search bar in blue can be used to search for a specific player or team. In the case of games, there is a start and end date filter. Double-clicking on a player, team, or game will give more information for whatever was selected.

Example 1: Let’s say the user wants to find the stats for a player whose last name is Henry. They know he plays for the Titans, but can’t remember his first name.

1. Click on the “Players” radio button if not already selected to navigate to the list of players.



1. In the search query, type “Henry” to filter players then click search. The resulting list should look like this:

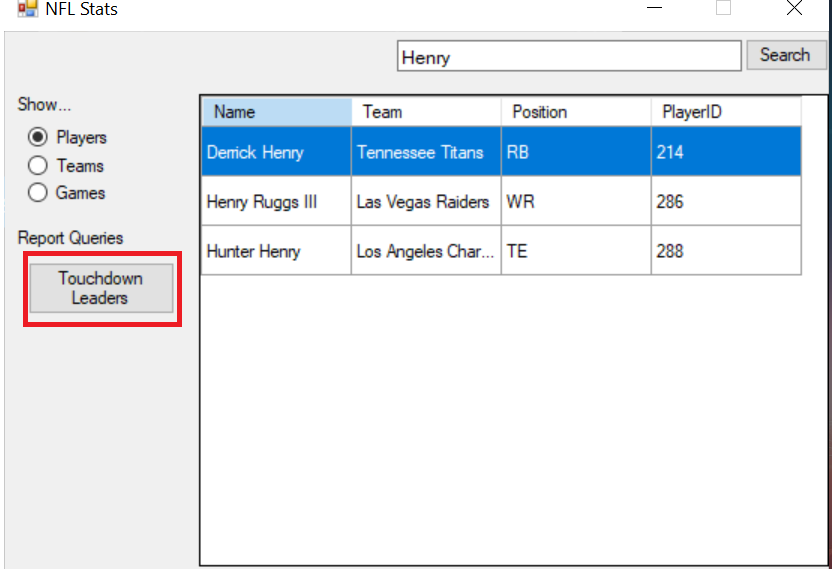


1. The user now recognizes Derrick Henry as the desired player. Double-clicking on that row will display the stats for him.

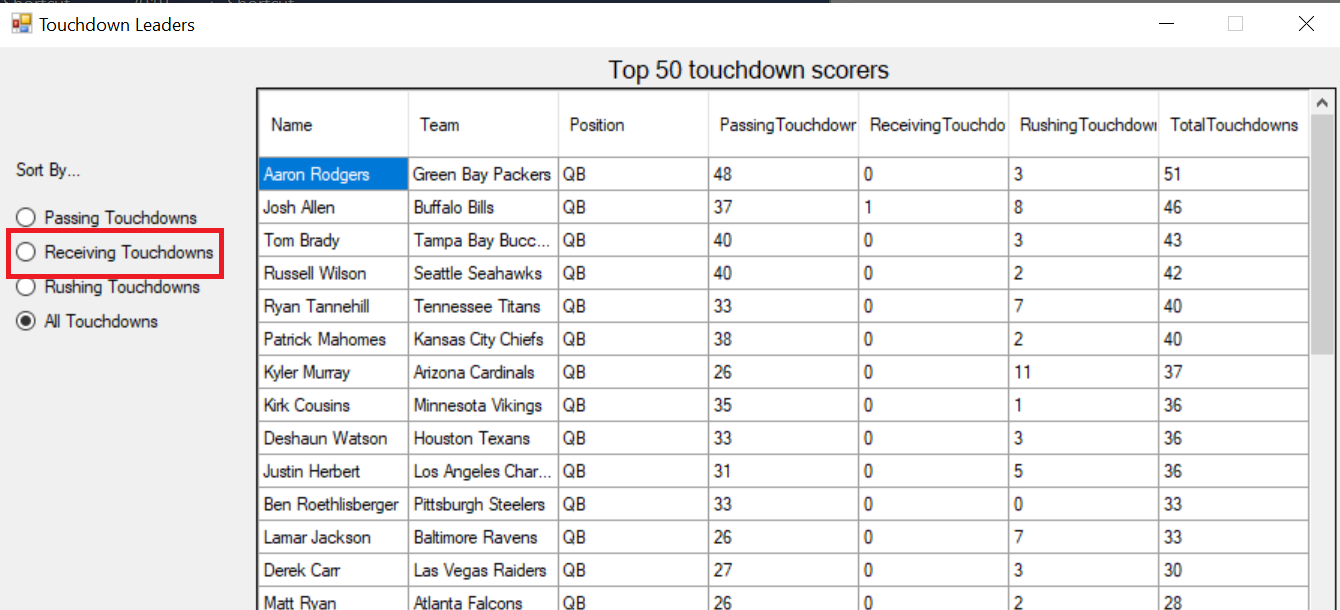


Example 2: The user wants to see the top players in terms of receiving touchdowns.

1. First click on the outlined button:



1. In the resulting window under “Sort By…”, choose “Receiving Touchdowns”:



**Report Queries:**

1. This report query takes the top 50 players based on touchdowns and ranks them. It can be ordered by either passing, receiving, rushing, or total given an order by parameter, but returns each as a column regardless.

Graphical user interface, text, application

Description automatically generated

When @OrderBy = ‘Receiving Touchdowns’ returns

Graphical user interface

Description automatically generated with medium confidence

1. This report query takes a game as a parameter and returns the total scrimmage yards for every player in that game by adding passing, receiving, and rushing.

Graphical user interface, text, application

Description automatically generated

When given @GameID = 66 returns

Table

Description automatically generated

1. The TeamTotals reports query takes a TeamID as a parameter and returns the total stats for that team by adding up every player’s stats for every game.Text, application

   Description automatically generated.

When given @TeamID = 4 returns



1. SearchGames returns a list of games given a range of dates.

Text

Description automatically generated with medium confidence

Given '2020-01-01' and '2021-01-01' returns

Table

Description automatically generated with medium confidence

**Summary And Discussion:**

* Our project is mostly what we had originally envisioned. Of course there were many small changes done in order to get it to work. A few of our report queries were changed as we didn’t see them fitting into our current design. We also put less focus on filtering on a specific stat and more on filtering for a specific player or team.
* The database design had minimal changes from the proposal. None of the relationships were changed, but there were some changes to the columns. The biggest change was taking out all the defensive stats from the PlayerGame table and adding more offensive stats. This was done due to difficulty in finding per game defensive stats data for each player. Abbreviation was added to the Team table and SourceGameID to the Game table in order to get a connection to the staging tables. Location in team was taken out as well as FirstName and LastName in Player to combine into just Name. Finally, location was removed from the Game table as this was redundant.
* The biggest thing we learned was connecting our database and being able to use the information in visual studio with our front end code. We also learned a lot about database design because of this, especially solving a multiple path problem which was a big issue we had to solve.
* There are a ton of improvements that could be made on this program. The most obvious is adding defensive stats and players. Also, right now top players can only be listed in terms of touchdowns, but this could also be done on every stat. Expansions such as fantasy points as well as a game predictor based on advanced stats could be added. Due to the abundance of information and data from the NFL, there’s basically unlimited possibilities.