**PROPOSAL by PETER GRANTCHAROV (pdg2116) and PO-CHIEH LIU (pl2441)**

HIGH LEVEL DESCRIPTION

Our application will:

- Provide a tool for users to query historic NBA games to get statistics of interest or view games of interest
- Let users place virtual bets (fake *money*, real performance) on NBA games for a given day
- Allow users to select variables of interest to generate estimated win probabilities for betting outcomes

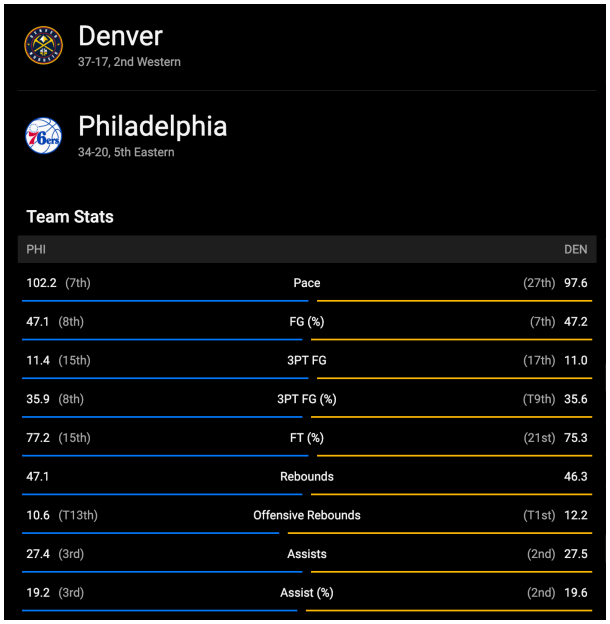
DATA SOURCES

The primary data sources for this project will be 1) historic NBA game box scores (statistics) scraped from *www.basketball-reference.com,* and 2) live sports book betting odds scraped from various sports books.

APPLICATION OVERVIEW

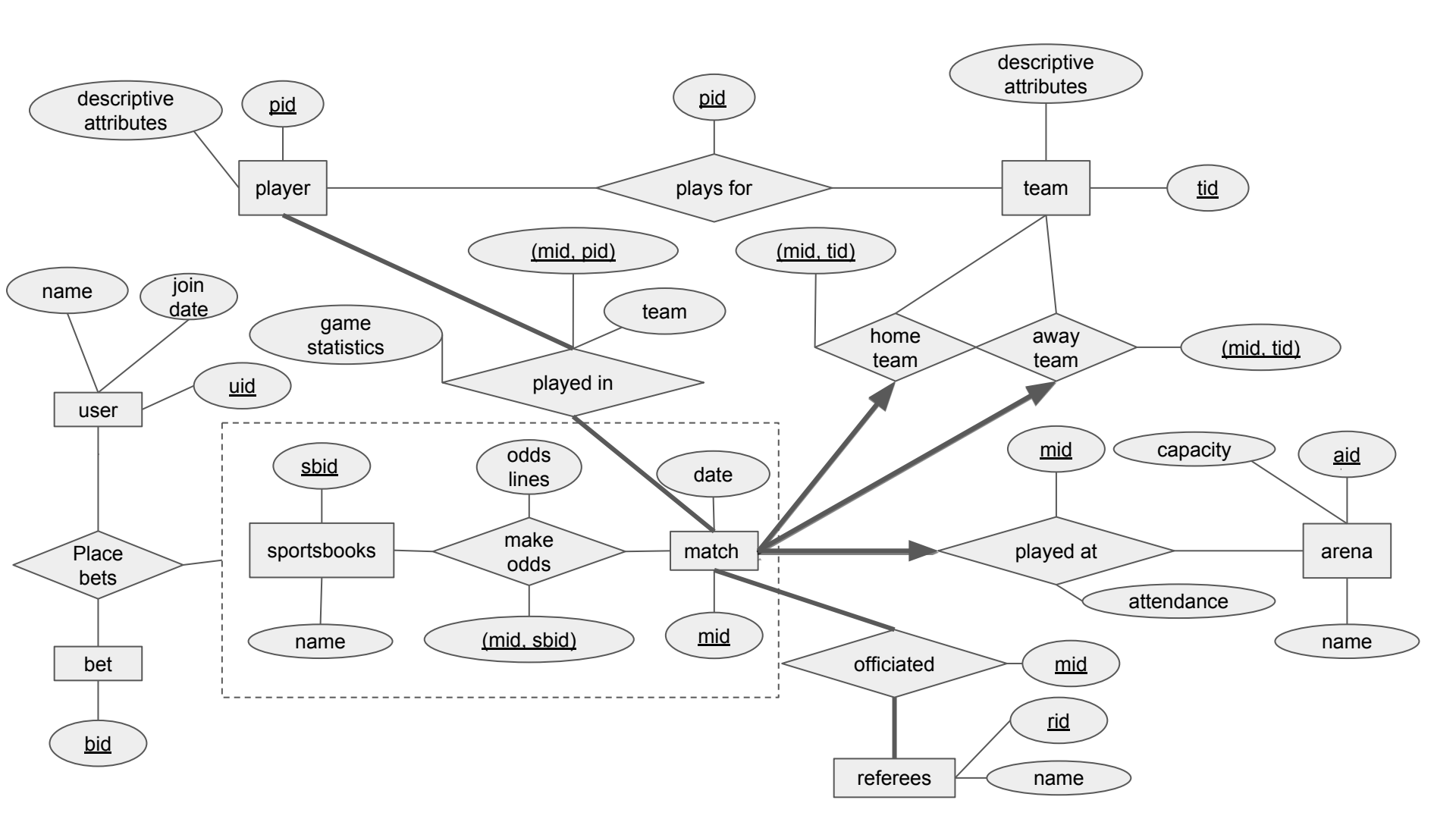
Our main page will contain all NBA games scheduled to be played on that day. For each match, the user will be able to click on a link to take them to a distinct page dedicated for this match.

On this "game page", there will be a section dedicated to displaying basic statistics for this match. This allows users to compare teams and will be automatically generated for a given matchup. Functionally, it will be very similar to the website screenshot shown.




Further, users will have a wide selection of querying options available to select the variables that they think are influential for outcome of this game. These variables include "home court advantage", "season records", "points per game", "record in last 10 games", etc. From these selections, we will query the database, display the statistics that the requested, and automatically insert the resulting statistics as features into a simple machine learning model to predict game outcomes. The user can then compare the output of their self-generated model to the betting lines that the sports books are offering, and finally, place bets themselves on the matches.

User betting results will be inserted into our database to allow for performance tracking over time, so that users will be able to see whether they truly have what it takes to be successful in the sports betting markets. Below is our ER-diagram for our application. but please note that due to the sheer amount of attributes, the diagram omits many *specific* names and constraints that are found in the SQL schema translation.


descriptive attributes
pid
player
pid
plays for
descriptive attributes
team
tid
(mid, pid)
team
played in
(mid, tid)
home team
away team
(mid, tid)
name
join date
user
uid
game statistics
sbid
sportsbooks
name
odds lines
make odds
(mid, sbid)
date
match
mid
Place bets
bet
bid
mid
capacity
aid
played at
arena
attendance
name
officiated
mid
rid
referees
name

SQL SCHEMA

```
-- stand alone tables
CREATE TABLE player (
     p_id int PRIMARY KEY,
     first_name text NOT NULL,
     last_name text NOT NULL,
     posit text NOT NULL,
     CHECK (posit = 'PG' OR
             posit = 'SG' OR
             posit = 'C' OR
             posit = 'PF' OR
             posit = 'SF'),
     shooting_hand text NOT NULL,
     CHECK (shooting_hand = 'L' OR shooting_hand = 'R')
);


CREATE TABLE team (
     t_id int PRIMARY KEY,
     name text NOT NULL,
     city text NOT NULL
);


CREATE TABLE arena (
     a_id int PRIMARY KEY,
     name text NOT NULL,
     capacity int NOT NULL,
     CHECK (capacity > 0)
);


CREATE TABLE users (
     u_id int PRIMARY KEY,
     name text NOT NULL
);


CREATE TABLE referee (
     r_id int PRIMARY KEY,
     name text NOT NULL
);
```

```
CREATE TABLE sportsbook (
     sb_id int PRIMARY KEY,
     name text NOT NULL
);


CREATE TABLE game(
     g_id int PRIMARY KEY,
    game_time timestamp NOT NULL,
     a_id int REFERENCES arena (a_id),
     t_id_home int REFERENCES team (t_id),
     t_id_away int REFERENCES team (t_id),
     CHECK (t_id_home != t_id_away),
     home_q1_score int NOT NULL,
     home_q2_score int NOT NULL,
     home_q3_score int NOT NULL,
     home_q4_score int NOT NULL,
     away_q1_score int NOT NULL,
     away_q2_score int NOT NULL,
     away_q3_score int NOT NULL,
     away_q4_score int NOT NULL,
     home_ot_score int NOT NULL,
     away_ot_score int NOT NULL,
     CHECK (
          home_q1_score >= 0 AND
          home_q2_score >= 0 AND
          home_q3_score >= 0 AND
          home_q4_score >= 0 AND
          away_q1_score >= 0 AND
          away_q2_score >= 0 AND
          away_q3_score >= 0 AND
          away_q4_score >= 0
          ),
     CHECK (
         home_ot_score >= 0 OR
         home_ot_score IS NULL
         ),
     CHECK (
         away_ot_score >= 0 OR
         away_ot_score IS NULL
         )
);
```

```
CREATE TABLE game_referee (
     g_id int REFERENCES game (g_id) ON DELETE CASCADE,
     r_id int REFERENCES referee (r_id),
     PRIMARY KEY (g_id, r_id)
);


CREATE TABLE player_game_stats (
     g_id int REFERENCES game (g_id) ON DELETE CASCADE,
     p_id int REFERENCES player (p_id),
     PRIMARY KEY (g_id, p_id),
     t_id int REFERENCES team (t_id),
     Minutes_played int NOT NULL,
     Field_goals_made int NOT NULL,
     Field_goal_attempts int NOT NULL,
     Three_pointers_made int NOT NULL,
     Three_point_attempts int NOT NULL,
     Free_throws_made int NOT NULL,
     Free_throw_attempts int NOT NULL,
     Offensive_rebounds int NOT NULL,
     Defensive_rebounds int NOT NULL,
     Assists int NOT NULL,
     Steals int NOT NULL,
     Blocks int NOT NULL,
     Turnovers int NOT NULL,
     Personal_fouls int NOT NULL,
     Points int NOT NULL,
     Plus_minus int NOT NULL,
     Offensive_rebound_percentage int NOT NULL,
     Defensive_rebound_percentage int NOT NULL,
     Total_rebound_percentage int NOT NULL,
     Assist_percentage int NOT NULL,
     Steal_percentage int NOT NULL,
     Block_percentage int NOT NULL,
     Turnover_percentage int NOT NULL,
     Usage_percentage int NOT NULL,
     Offensive_rating int NOT NULL,
     Defensive_rating int NOT NULL,
     CHECK (
         Field_goals_made <= Field_goal_attempts AND
         Three_pointers_made <= Three_point_attempts AND
         Free_throws_made <= Free_throw_attempts
         ),
     CHECK (
         Minutes_played >= 0 AND
         Field_goals_made >= 0 AND
```

```
          Field_goal_attempts >= 0 AND
          Three_pointers_made >= 0 AND
          Three_point_attempts >= 0 AND
          Free_throws_made >= 0 AND
          Free_throw_attempts >= 0 AND
          Offensive_rebounds >= 0 AND
          Defensive_rebounds >= 0 AND
          Steals >= 0 AND
          Blocks >= 0 AND
          Turnovers >= 0 AND
          Personal_fouls >= 0 AND
          Points >= 0 AND
          Offensive_rebound_percentage >= 0 AND
          Defensive_rebound_percentage >= 0 AND
          Total_rebound_percentage >= 0 AND
          Assist_percentage >= 0 AND
          Steal_percentage >= 0 AND
         Block_percentage >= 0 AND
          Turnover_percentage >= 0 AND
          Usage_percentage >= 0 AND
          Offensive_rating >= 0 AND
          Defensive_rating >= 0
          )
);

-- sportsbook table
CREATE TABLE sb_game_odds(
     g_id int REFERENCES game (g_id) ON DELETE CASCADE,
     sb_id int REFERENCES sportsbook (sb_id) ON DELETE CASCADE,
    odds_time timestamp, -- sportsbook will update odds
intermittently
     PRIMARY KEY (g_id, sb_id, odds_time),
     h_money_line int,
     a_money_line int,
     h_spread int,
     a_spread int,
     over_line int,
     under_line int,
     spread_value int,
     over_under_value int
);

-- user bet table
CREATE TABLE place_bet(
     b_id int PRIMARY KEY,
     u_id int REFERENCES users (u_id) ON DELETE CASCADE,
```

```
    g_id int REFERENCES game (g_id) ON DELETE CASCADE,
    sb_id int REFERENCES sportsbook (sb_id) ON DELETE CASCADE,
    bet_time timestamp NOT NULL,
    bet_h_money_line int,
    bet_a_money_line int,
    bet_h_spread int,
    bet_a_spread int,
    bet_over_line int,
    bet_under_line int,
    bet_spread_value int,
    bet_over_under_value int
);
```