NBA Draft Simulation

Ishaan Lodhi
Saketh Kilaru
Luis Riviere
Akshaj Salvi

IST659:
Data Administration Concepts &
Database Management

Mark Romano

December 12th, 2024

NBA Draft Simulation: An SQL-Powered Bidding Interface

Introduction

For this project we decided to simulate an NBA player bidding marketplace in SQL, pushing it in an unconventional direction. In this mock draft simulation system, we changed the rules from the real-world NBA draft to introduce bidding, making the process more interesting by having teams compete for the same players while maintaining fairness typically seen in US sports draft. The goal of this project is to highlight SQLs ability to simulate complicated real-world behavior. Software and apps such as the one we developed can be used to forecast bids and adjust strategies.

Business Rules and Requirements

- The bidding process involves 60 players and 20 teams, divided into 4 lots, with each lot comprising 5 players and 5 teams (e.g., Lot 1 includes players ranked 1–5 and teams ranked 16–20).
- Bidding occurs lot by lot, ensuring all players and teams in a lot complete their transactions before progressing to the next.
- Each team can bid for a player only if the player's position matches one of the team's three preferred positions.
- If no match exists, the player is automatically assigned to the lowest-ranked team within the lot that has not yet acquired a player, deducting 50% of the team's live budget during the first two rounds and 100% in the third round.
- During Rounds 1 and 2, teams can bid up to 50% of their live budget, while in Round 3, they can use their entire remaining budget.
- A team that successfully acquires a player within a lot is frozen out from further bidding for that lot, ensuring each team acquires exactly one player per lot. This process ensures that at the end of each lot, all players are assigned to exactly one team, allowing the next lot to begin.

Entities and Attributes:

1. **Teams:**
   - team_id (primary key)
   - team_name (unique)
   - team_total_budget
   - team_live_budget
   - team_ranking (unique)
   - team_player_count
   - team_lot_round_1
   - team_lot_round_2
   - team_lot_round_3

2. **Players:**
   - player_id (primary key)
   - player_firstname
   - player_lastname
   - player_ppg (points per game)
   - player_apg (assists per game)
   - player_rpg (rebounds per game)
   - player_impact
   - player_position
   - player_ranking (unique)
   - player_current_lot

3. **Bids:**
   - bid_id (primary key)
   - bid_by (foreign key referencing Teams)
   - bid_for (foreign key referencing Players)
   - bid_amount
   - bid_status

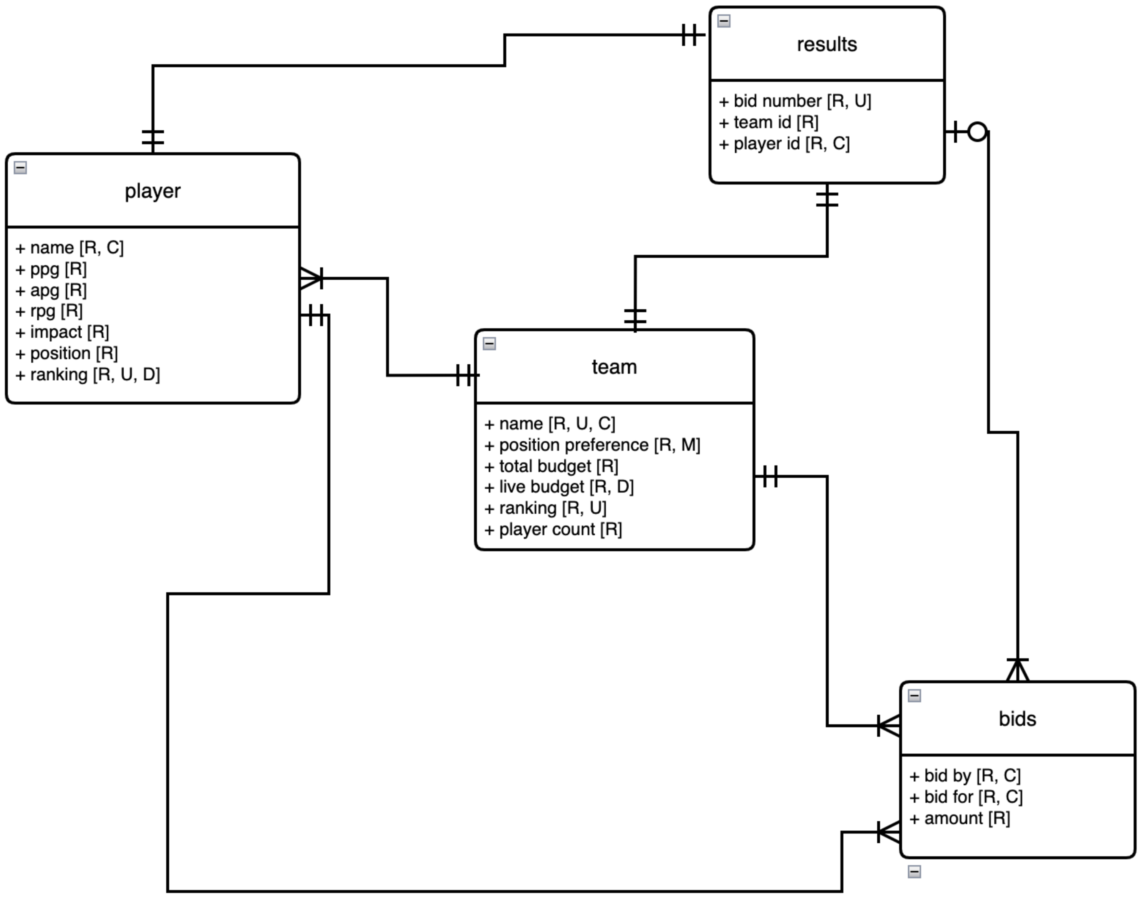
4. **Preferences:**
   - preference_id (primary key)
   - preference_team_id
   - preferred_player_position_1
   - preferred_player_position_2
   - preferred_player_position_3

5. **Results:**

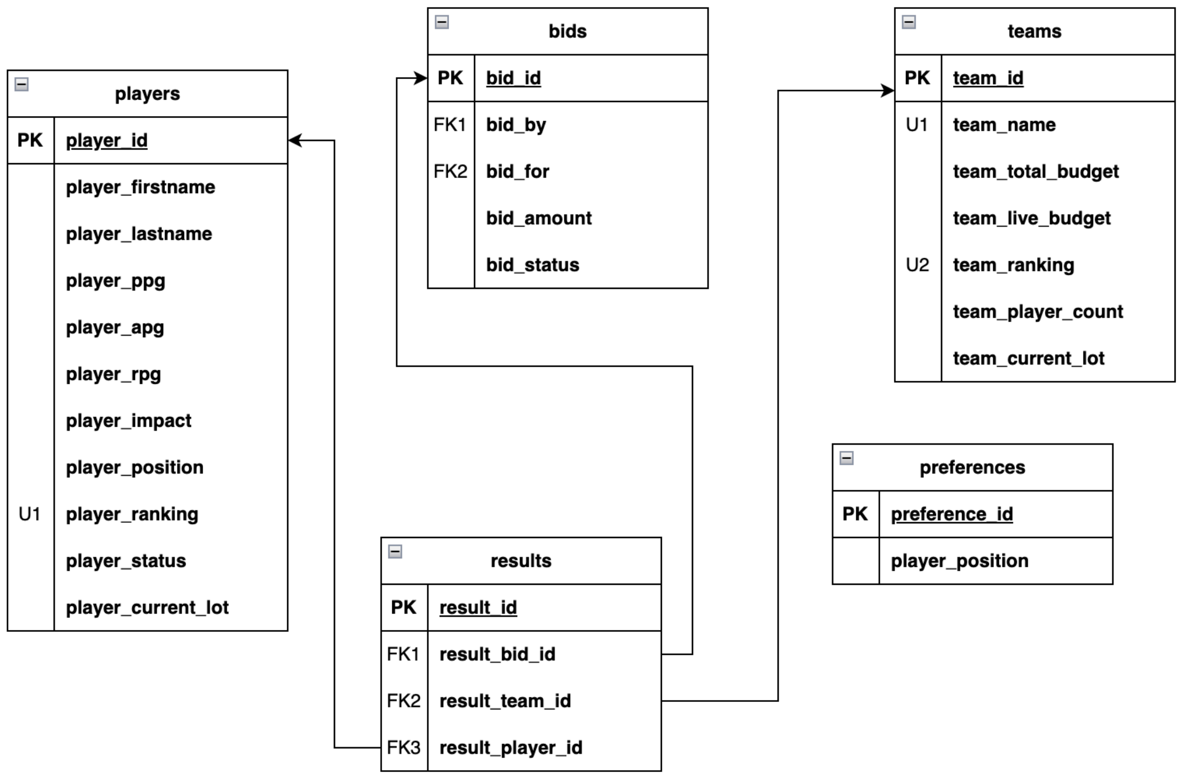
- result_id (primary key)
- result_bid_id (foreign key referencing Bids)
- result_team_id (foreign key referencing Teams)
- result_player_id (foreign key referencing Players)

Looking at the Business Rules and the entities, it is evident that the architecture of the entities and relationships simplify the application of business rules. Every table is connected to the others with the help of foreign keys, and redundancy has been kept at an absolute minimum.

Crow’s Foot Diagram

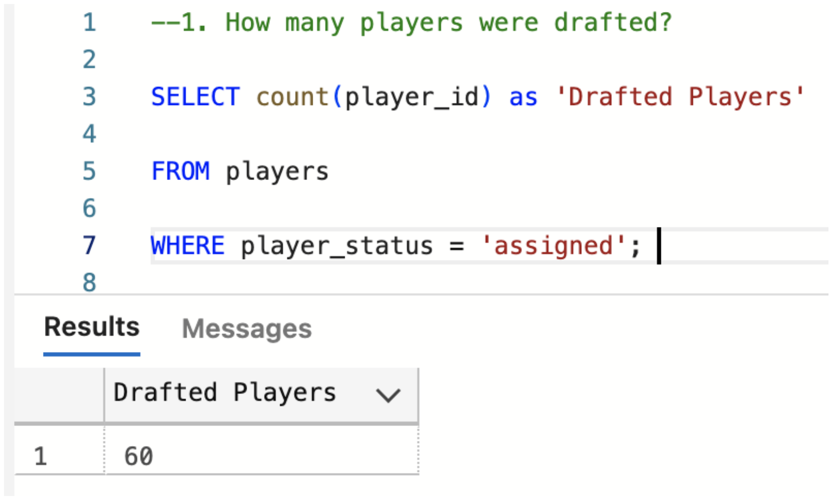


Logical Model Diagram



Questions and Queries

1. Which players have been drafted?

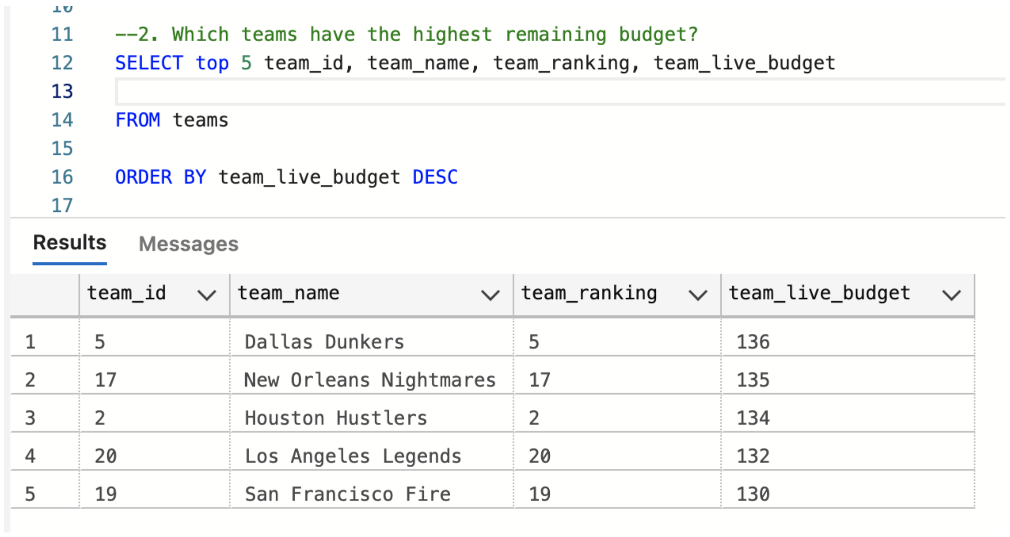


```sql
--1. How many players were drafted?

SELECT count(player_id) as 'Drafted Players'

FROM players

WHERE player_status = 'assigned';

```

Results Messages

	Drafted Players
1	60

2. Which teams have the highest remaining budget?

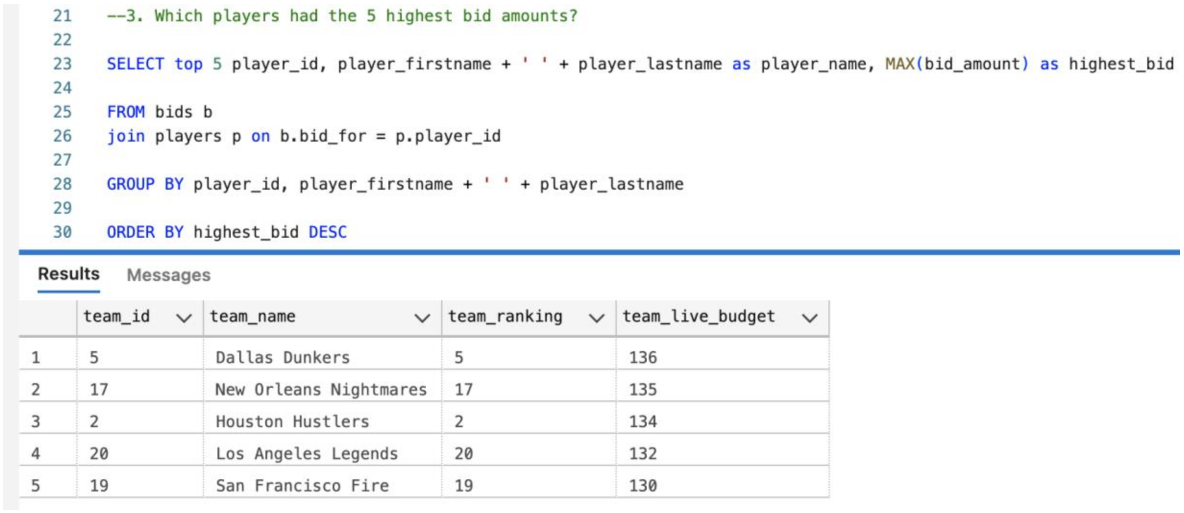


```sql
--2. Which teams have the highest remaining budget?
SELECT top 5 team_id, team_name, team_ranking, team_live_budget

FROM teams

ORDER BY team_live_budget DESC

```

Results Messages

	team_id	team_name	team_ranking	team_live_budget
1	5	Dallas Dunkers	5	136
2	17	New Orleans Nightmares	17	135
3	2	Houston Hustlers	2	134
4	20	Los Angeles Legends	20	132
5	19	San Francisco Fire	19	130

3. Which players had the highest bid amounts?

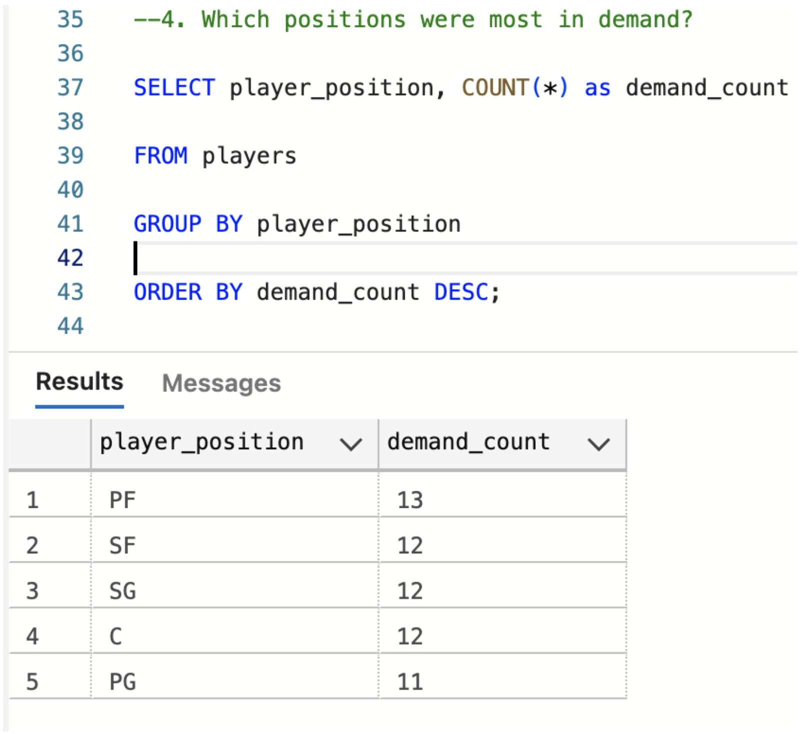


```
--3. Which players had the 5 highest bid amounts?

SELECT top 5 player_id, player_firstname + ' ' + player_lastname as player_name, MAX(bid_amount) as highest_bid

FROM bids b
join players p on b.bid_for = p.player_id

GROUP BY player_id, player_firstname + ' ' + player_lastname

ORDER BY highest_bid DESC
```

Results   Messages

	team_id	team_name	team_ranking	team_live_budget
1	5	Dallas Dunkers	5	136
2	17	New Orleans Nightmares	17	135
3	2	Houston Hustlers	2	134
4	20	Los Angeles Legends	20	132
5	19	San Francisco Fire	19	130

4. Which positions were most in demand?

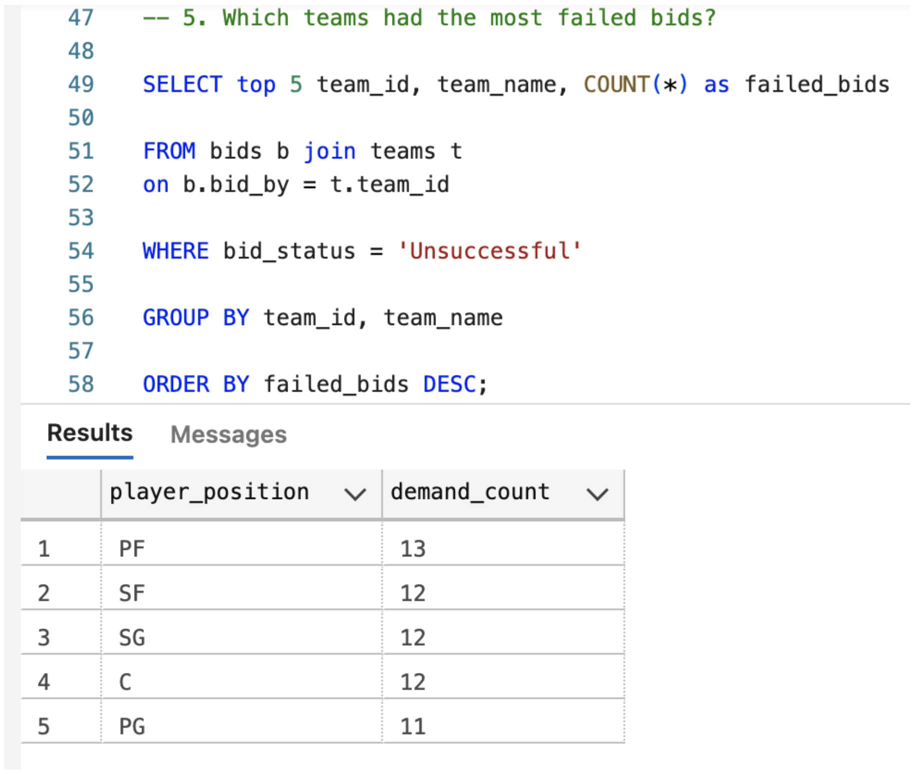


```
--4. Which positions were most in demand?

SELECT player_position, COUNT(*) as demand_count

FROM players

GROUP BY player_position

ORDER BY demand_count DESC;
```

Results   Messages

	player_position	demand_count
1	PF	13
2	SF	12
3	SG	12
4	C	12
5	PG	11

5. Which teams had the most failed bids?

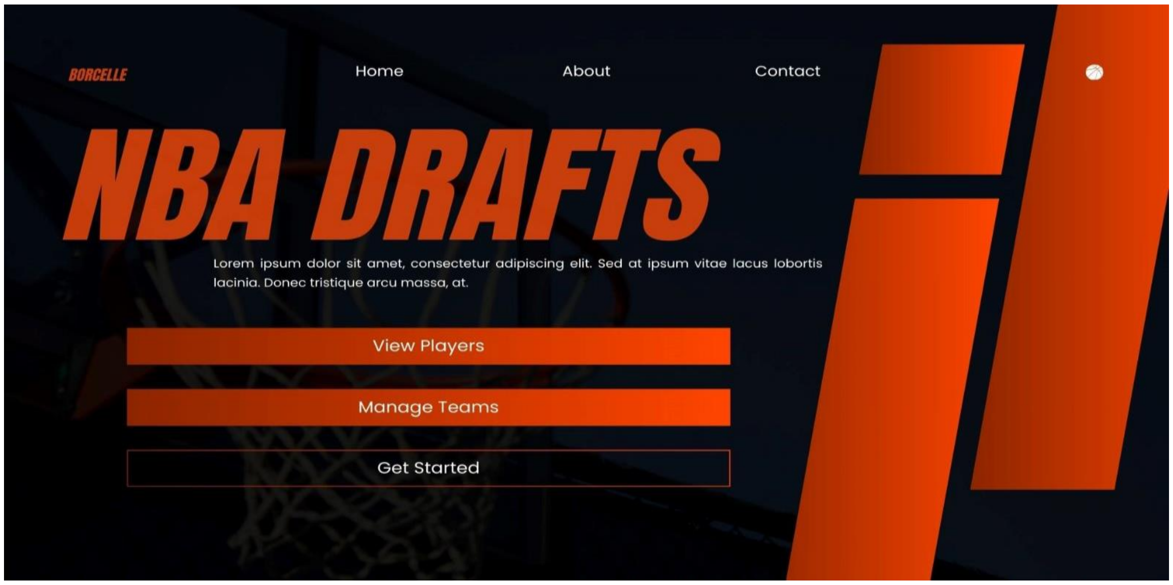


```sql
-- 5. Which teams had the most failed bids?

SELECT top 5 team_id, team_name, COUNT(*) as failed_bids

FROM bids b join teams t
on b.bid_by = t.team_id

WHERE bid_status = 'Unsuccessful'

GROUP BY team_id, team_name

ORDER BY failed_bids DESC;
```

Results Messages

	player_position	demand_count
1	PF	13
2	SF	12
3	SG	12
4	C	12
5	PG	11

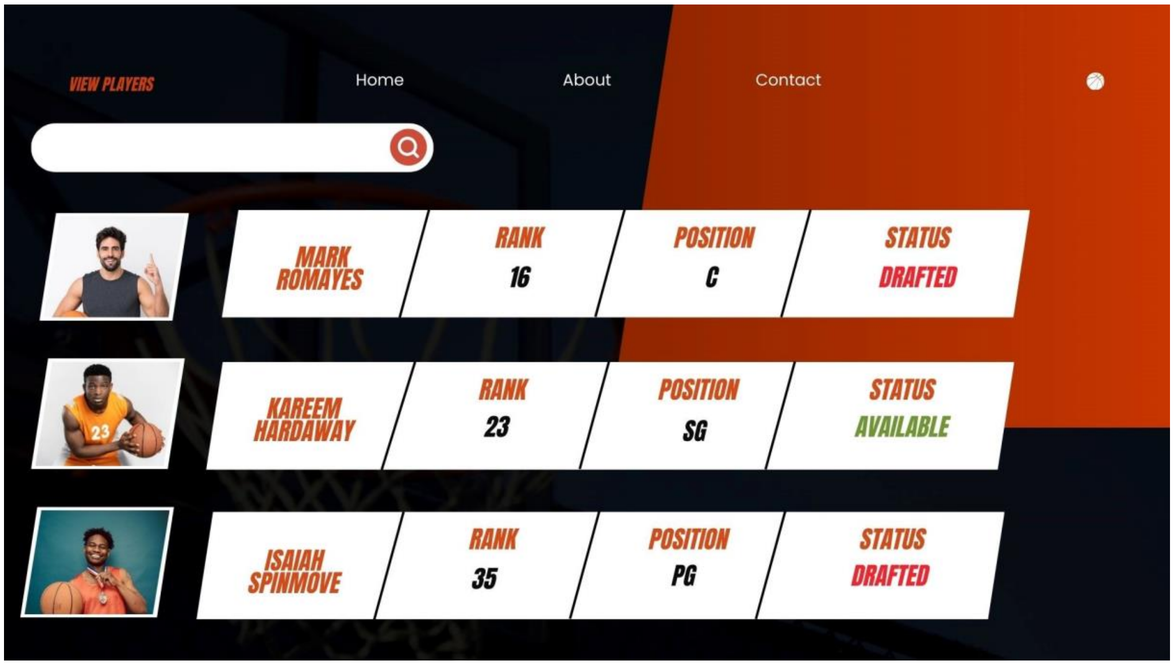
User Diagram



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ipsum vitae lacus lobortis lacinia. Donec tristique arcu massa, at.

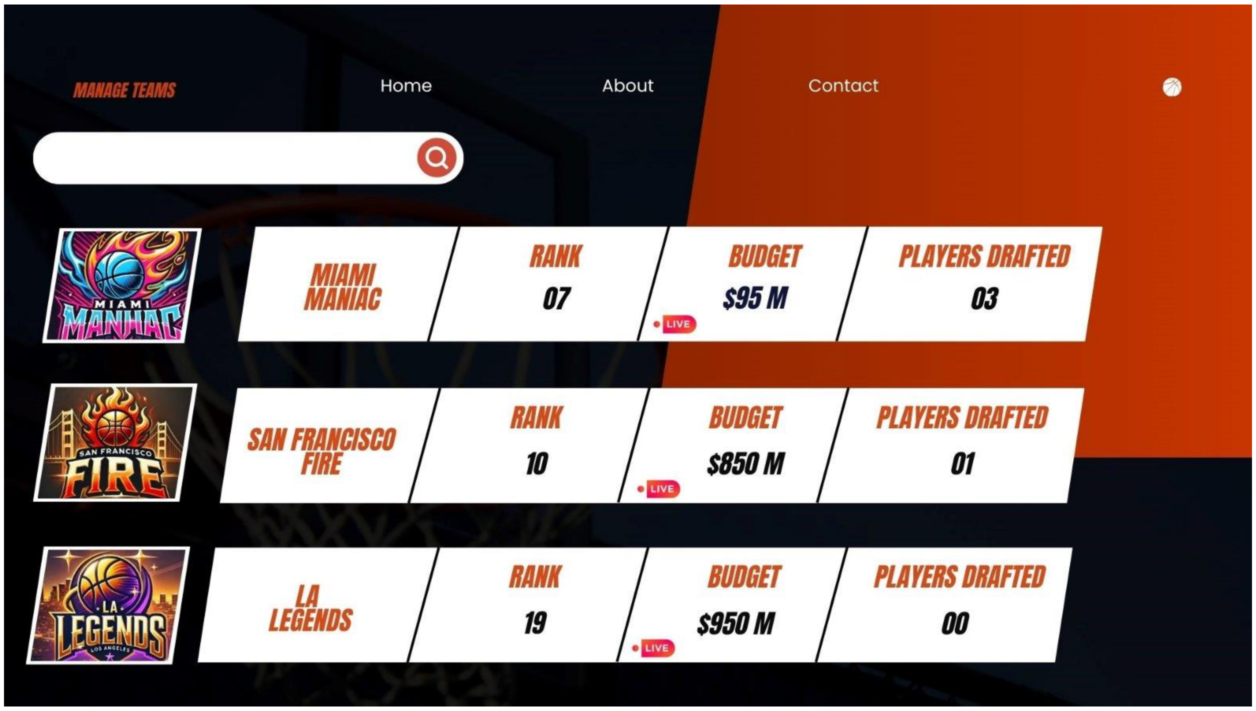
This is the landing page where users can view players and their status, manage their fantasy team. If you login as a team, you will be provided a sandbox environment for simulating the bidding process.

Example of the view players tab. Teams can see player rank, their position, and their availability to be drafted.



MARK ROMAYES	RANK 16	POSITION C	STATUS DRAFTED
KAREEM HARDAWAY	RANK 23	POSITION SG	STATUS AVAILABLE
ISAIAH SPINMOVE	RANK 35	POSITION PG	STATUS DRAFTED

This page provides information about players. Players that have recently been drafted also appear here to provide teams and users with updated information. General player information such as their rank and position appear by default.



	RANK	BUDGET	PLAYERS DRAFTED
MIAMI MANIAC	07	$95 M LIVE	03
SAN FRANCISCO FIRE	10	$850 M LIVE	01
LA LEGENDS	19	$950 M LIVE	00

Teams can see other teams' rank, budget and how many players they have drafted. This helps the teams dynamically strategize for players, positions and their budget.

Bidding Simulation

Draft Sequence:

- Conduct drafts in sequential lots.
- Assign all players in a lot before moving to the next.

Team Eligibility Evaluation:

- Consider team budgets and constraints.
- Facilitate competitive bidding based on player ranking and availability.

Bidding Guidelines:

- Early Rounds: Limit bids to 50% of remaining budget.
- Later Rounds: Allow full budget flexibility.

Player Assignment Process:

- Highest bidder signs the player.
- Deduct bid amount from the team's live budget.
- Update the team's player count.
- Mark the player’s status as "assigned."

Handling Unsuccessful Bids:

- Move to the next player if no valid bids are placed.

Draft Results Recording:

- Document all results of each draft.
- Include both successful and unsuccessful bids.
- Log player allocations to teams for transparency.
- **Fair and Efficient Allocation:**
- Ensure rule-based and efficient player allocation.
- Maintain compliance with draft rules.

Reflection/Moving in the Future

This project was an incredible learning experience, as it required us to dive deep into both the technical and strategic aspects of simulating an NBA draft auction. While we achieved many of our initial goals, there are several areas where, given more time, resources, and experience, we could enhance the functionality and make the system more dynamic and engaging. Here are some key reflections and ideas for improvement:

1. Implementing a Loser's Bracket

One thing we could have done differently is how teams handle situations where none of their preferred positions are available. Currently, such teams are automatically assigned the highest-ranked available player, which can sometimes feel forced and less strategic. Instead, we could implement a **loser's bracket**, where these teams wait until the draft is completed and then select from the undrafted players based on their preferred positions. This change would:

- Give teams more control over their roster-building process.
- Add a layer of strategy to the system, as teams would need to weigh the risks of waiting for the loser's bracket versus bidding aggressively during the main draft.

2. Introducing Trades Between Teams

Allowing trades would significantly enhance the realism and strategic depth of the draft. Teams could trade players or draft picks if they find a mutually beneficial deal. For example:

- A team that drafts a highly ranked player in a position they don't need could trade that player for someone who better fits their requirements.
- A team with surplus budget could trade it for a draft pick or a higher-ranked player. This feature would simulate real-world scenarios where teams negotiate and adapt during drafts, making the app feel more interactive and engaging.

3. Dynamic Budgeting

Our current system requires teams to spend 50% of their live budget for the first two rounds, which simplifies budget management but reduces flexibility. Instead, we could allow teams to dynamically allocate their budget based on how much they value a particular player. For example:

- Teams could bid aggressively for a player they truly need and spend conservatively in other rounds.
- This approach would reflect the real-world decision-making process, where teams don't always follow rigid spending rules but adjust based on priorities and opportunities.

4. Customizable Team Preferences

Currently, team preferences (e.g., position priorities, budgets) are predefined. A more dynamic system would let teams define their own preferences before the draft begins. For example:

- Teams could prioritize a specific position, like point guards, based on their roster needs.
- They could also choose their budgeting strategy, such as saving for later rounds or focusing heavily on the first round. This customization would increase user engagement and make the simulation feel tailored to individual team strategies.

5. Advanced Player Ranking System

Player rankings in the current system are static. To make the system more realistic, we could base rankings on key performance stats like points per game (PPG), rebounds per game (RPG), assists per game (APG), and plus/minus (+/-). This would:

- Provide a more data-driven and transparent way to rank players.
- Allow users to analyze player stats and make informed decisions. For instance, teams could prioritize players with higher efficiency ratings over those with just scoring prowess, mimicking real-world scouting processes.

6. Adding Game Theory Dynamics

With more time, resources, and advanced knowledge in our MS program, one area we'd find highly insightful to explore is **integrating game theory elements** into the system. This would involve programming teams to predict the behaviors and strategies of other teams and adjust their bids accordingly. For example:

- Teams could analyze trends, such as which positions other teams are likely to bid for, and modify their own strategy to either compete or pivot to alternative players.
- Teams could recognize patterns in budgeting strategies (e.g., aggressive early bidding vs. conservative late bidding) and counteract those strategies to maximize their own outcomes.
- This would elevate the simulation from a static process to a dynamic environment where every decision influences the behavior of others, making it closer to the complexities of real-world NBA drafts.

Lessons Learned

Through this project, we learned the importance of balancing technical complexity with user experience. While we implemented a functional and engaging system, we realized that features like trading, customization, and game theory could significantly improve the app's usability and realism. Additionally, handling edge cases like undrafted players or budget mismanagement required careful planning and creativity.

Meeting Log

Description when you are meeting how you are sharing files

Meeting Date	In Attendance	Discussion Points
10/03/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	Came up with 2 ideas: NBA mock draft and mental health clinic network.
10/07/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	Finalised on the NBA mock draft. Came up with a few tables: Player table with stats; team (20) lookup; teams with budget, preferred position(s); position lookup, team rank; bids; you cannot bid for a position you're not in for; only teams ranked 15th and lower can bid for the first 5 players; a team can look for 3 positions max; add a min first bid; randomize the budget; 50%-25%-25% bids; player rankings;
10/10/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	Started drafting the conceptual model. How do we implement the preferences constraint for each team? Of the 3 preferred positions, 1 is a misc card. It will only be used if the other 2 are not available. Loser's bracket: If a team is unable to get their first 2 preferences after they've used their misc, all of them go to a loser's bracket where a rebidding happens.
10/17/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	Ran a simulation draft Questions to ask prof: Give us validation on our quantity of tables. Make the database rank the player using stats Ran a mock bid using chatGPT and verified our constraints.
10/21/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	players: names, position, stats, player id teams: name, team id, required position, total budget, live budget (live - amount), ranking, number of players drafted bids: bid_by (team_id fk), bid_for (player_id fk), amount, 60 players ONLY. 3 players per team; 20 teams Constraints: **if players_drafted == 0, bid_amount = 50% of live budget** **if players_drafted == 1, bid_amount = 50% of live budget** **if players_drafted == 2, bid_amount = 50% of live budget**
10/24/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	we need to have a bid ranking (in terms of amount) jotted down the ER data requirements. The bidding will happen in lots. The last 5 teams HAVE TO draft the first 5 players. Only then will the next lot start.

11/18/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	We are adding a separate view table for the results as we are using team_id as the FK and not team_name Also we need to add a constraint that a bid cannot be negative
12/2/2024	Ishaan Lodhi, Saketh Kilaru, Luis Riviere, Akshaj Salvi	live budget only changes if bid is successful (trigger/stored procedure) the bid_amount constraint (50% of current amount) should only be valid for the first 2 successful bids. Figure out how to write and enforce this constraint. live_bid_counter as a possible column? once a player is bought, he cannot be bid on again. player_status as a possible column? only 5 players are bid on at a time only the last 5 ranked teams can pick the first 5 ranked players teams look at the 5 player batch and look for their preferred position if its there they lock in their choice and give up other bids. Positions > ranking. Bid is 50% of live budget or rest of budget if third bid.