Team Deckard

Parker Morrissey, Hamza Ouboukhlik, Mackenzie Laurisden, and Alex Xayachanla

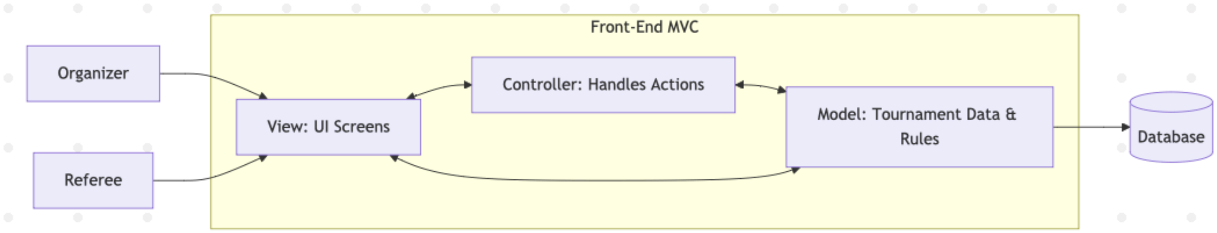
Initial Design

Purpose:

The system under design is an application that will provide an easy and efficient way to create and monitor tournament events of various sports. The main users of this application will primarily be tournament organizers.

System Structure:

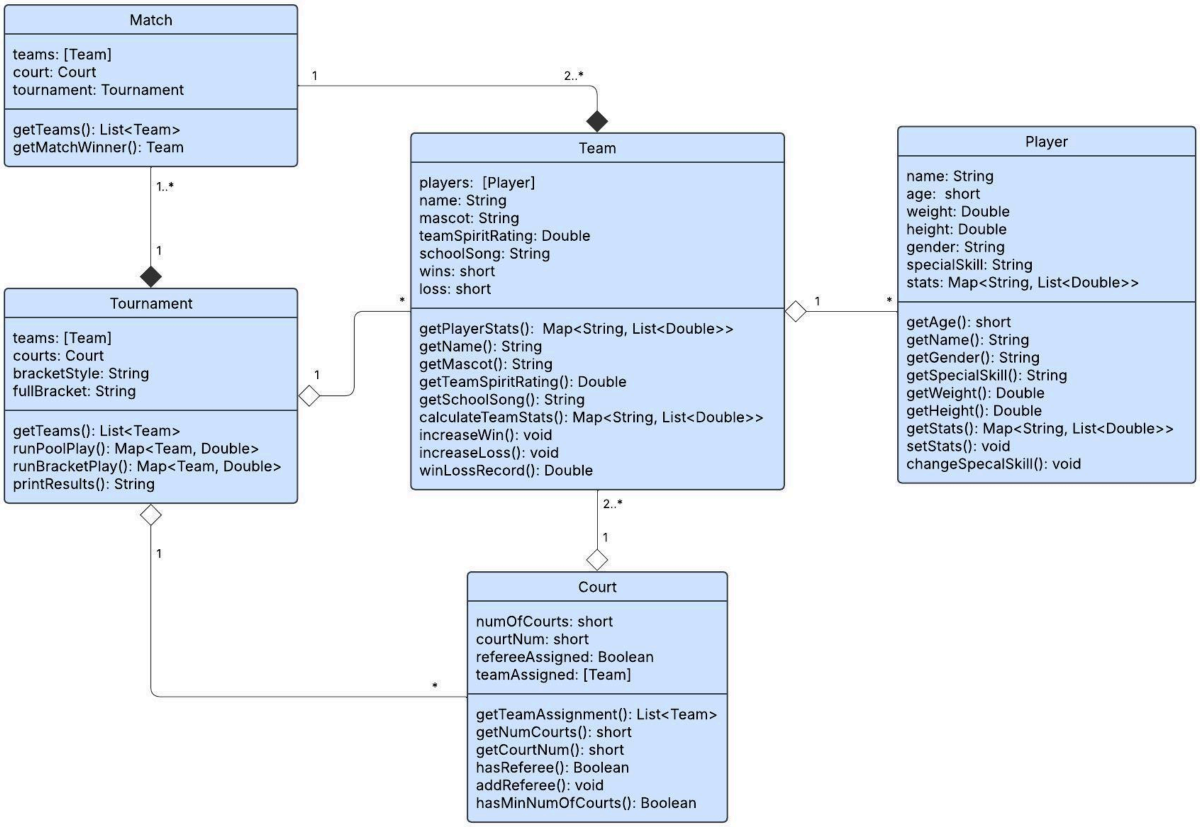
A model-view-controller will be used for this application. The model itself will store all of the team information and overall structure of the tournament, the view will display the user-interface where the user will be able to interact with various buttons to display team information, player stats, etc., and the controller will be used to actually implement tournament settings which the user will input via the view model.



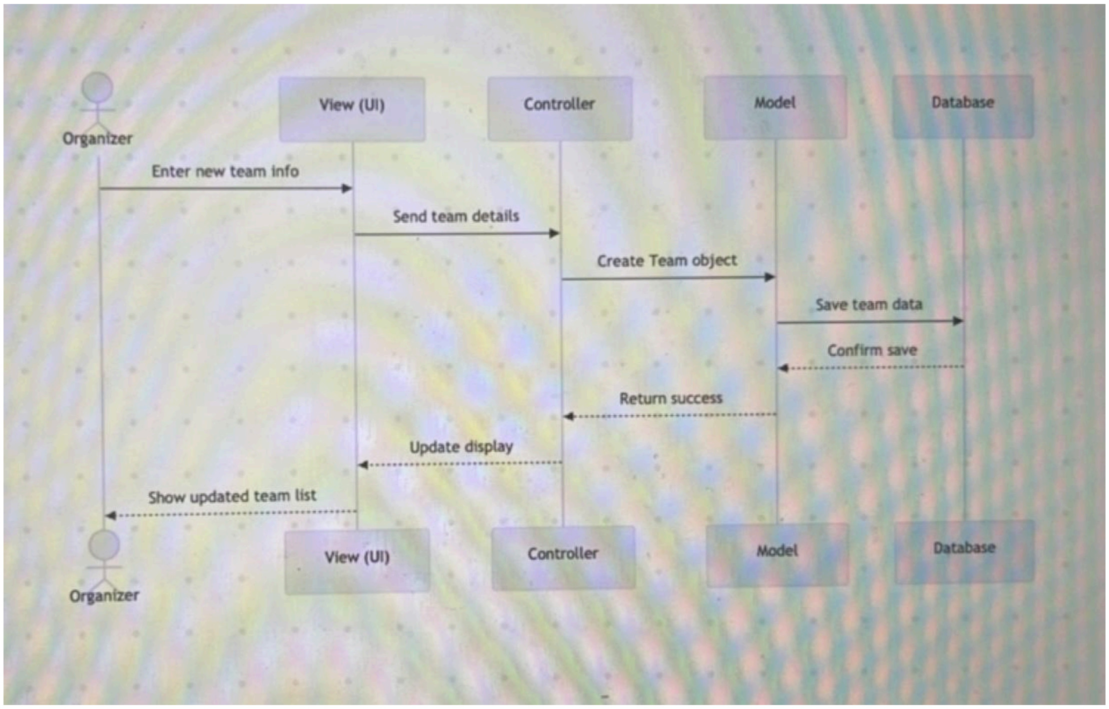
Class Design:

List of classes:

1. Team - This will hold a list of Players and will contain calculated team stats. Also includes the name of the team and eventually a mascot to uniquely identify the team. The team will also hold its win/loss record.
2. Player - Individual players on each team that will hold personal stats. Will also include the player's name, age, gender, weight, height and eventually the type of "special skills" they have (ie. "screaming at the ref.")
3. Tournament - Will hold teams. Also will hold the tournament results that will be displayed to the user, which will showcase the pool play results, bracket play results, overall placements and stats of each team that competed
4. Court - Holds which teams are currently playing on it.
5. Match - Will hold the teams that are competing in the current game. Will also show which court the game is on. It will also dictate the winner and use the Team class to calculate the stats from the game. Finally, the class will use the Tournament class to update pool play/bracket play.

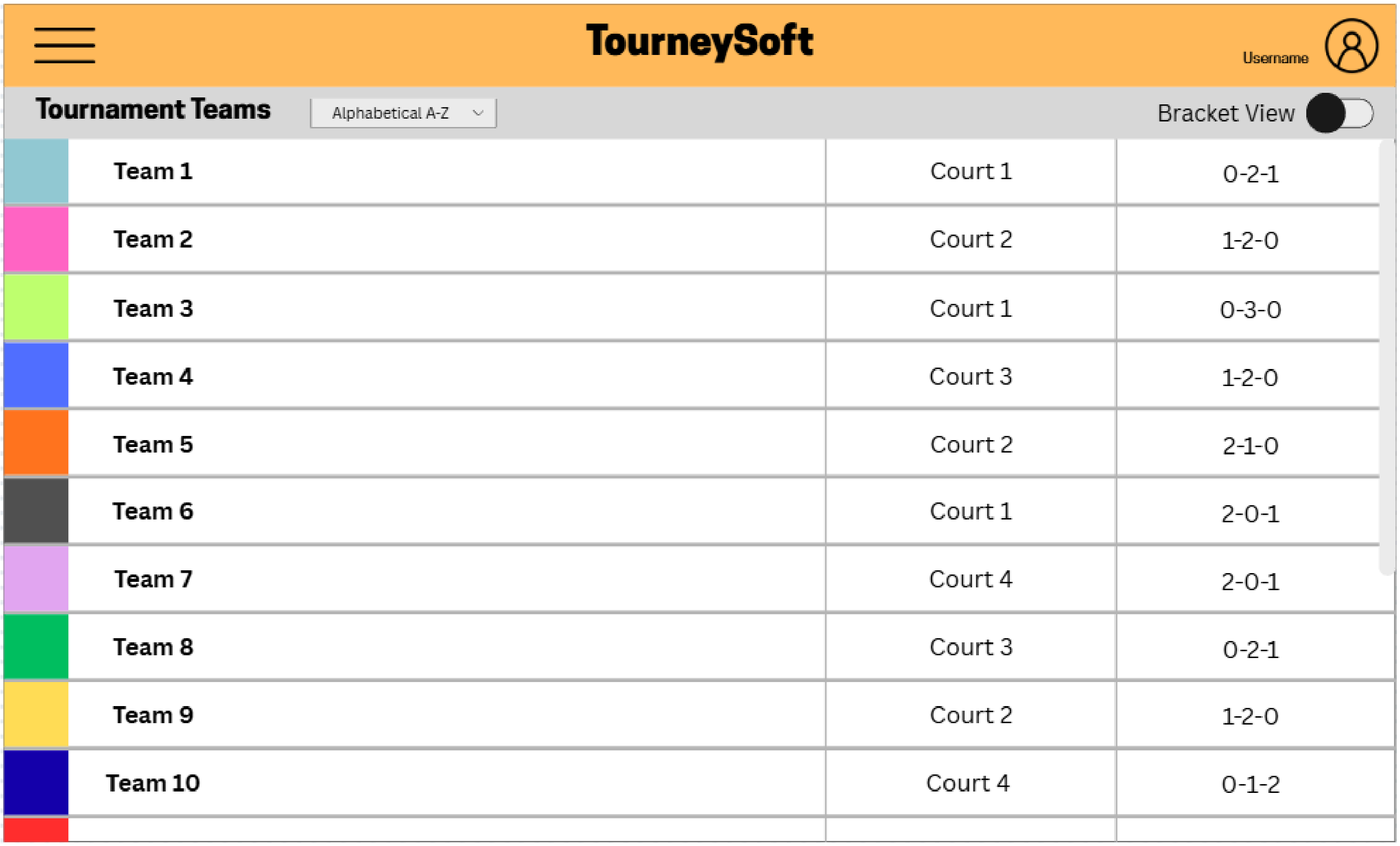


Sequence Diagram:

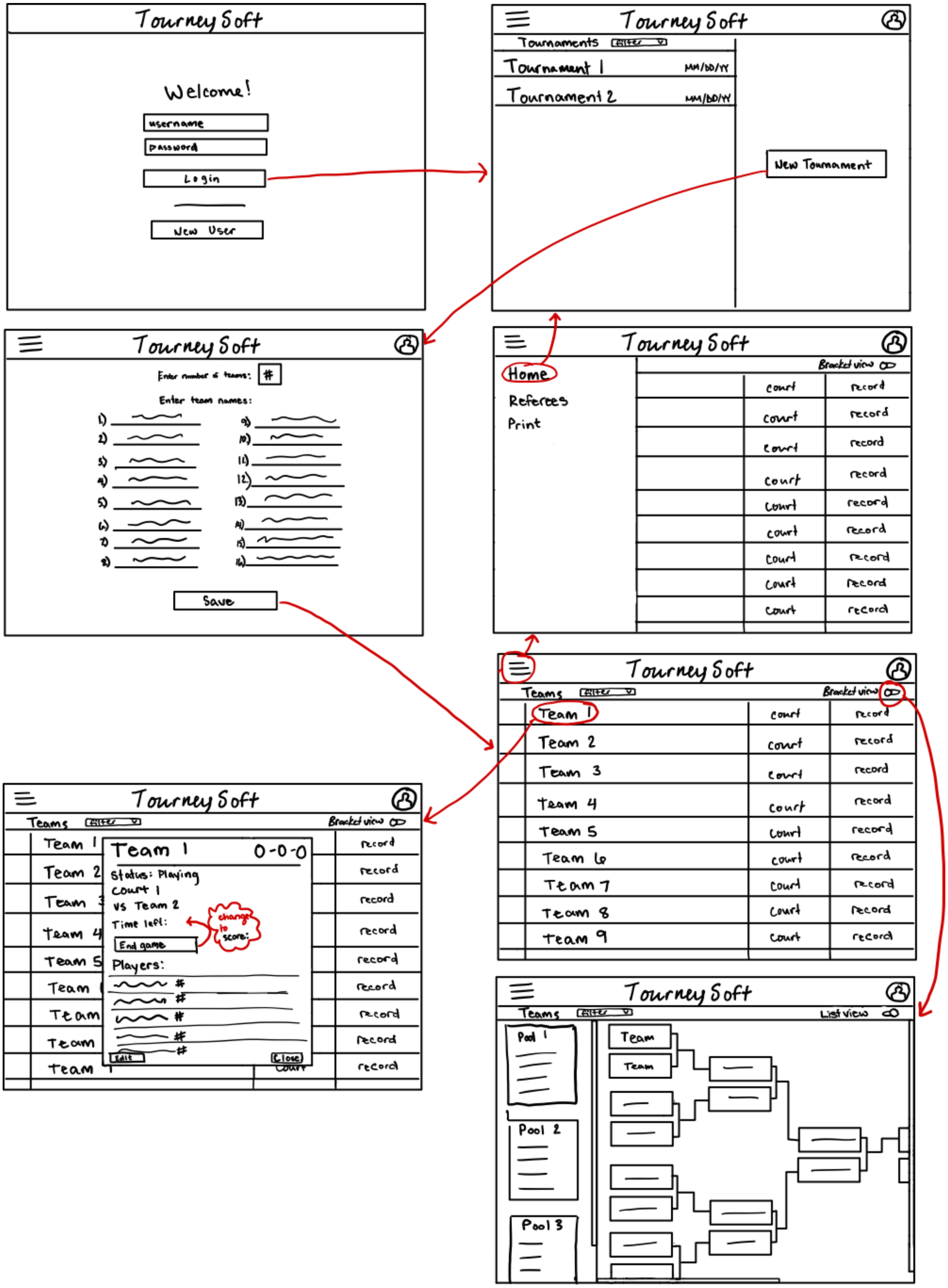


UI Diagram and Wireframe:

This UI mockup shows the screen for viewing the teams, in a list format. Each team is listed with their current court assignment, current rank, and a place to insert a team image (the colored boxes). It includes a toggle for changing the view to show the bracket, as well as the ability to change how the teams are listed (i.e alphabetical, rank, etc). Each team name will be clickable, and open a pop-up window with more information about the team. The orange banner will be viewable from every screen in the application and displays the software name, the current user of the system, and a hamburger menu option. The hamburger menu will include options for home, printing results, and a referee specific tab.



🟦	Team 1	Court 1	0-2-1
🟪	Team 2	Court 2	1-2-0
🟩	Team 3	Court 1	0-3-0
🟦	Team 4	Court 3	1-2-0
🟧	Team 5	Court 2	2-1-0
⬛	Team 6	Court 1	2-0-1
🟪	Team 7	Court 4	2-0-1
🟩	Team 8	Court 3	0-2-1
🟨	Team 9	Court 2	1-2-0
🟦	Team 10	Court 4	0-1-2
🟥			


Tourney Soft
Welcome!
username
password
Login
New User
Tourney Soft
Tournaments
Filter
Tournament 1 MM/DD/YY
Tournament 2 MM/DD/YY
New Tournament
Tourney Soft
Enter number of teams: #
Enter team names:
1)
2)
3)
4)
5)
6)
7)
8)
9)
10)
11)
12)
13)
14)
15)
16)
Save
Tourney Soft
Bracket view
Home
Referees
Print
court
record
Tourney Soft
Teams
Filter
Bracket view
Team 1
Team 2
Team 3
Team 4
Team 5
Team 6
Team 7
Team 8
Team 9
court
record
Tourney Soft
Teams
Filter
Bracket view
Team 1
Team 2
Team 3
Team 4
Team 5
Team
Team
Team
Team
0-0-0
Status: Playing
court 1
vs Team 2
Time left:
End game
change to score:
Players:
#
Edit
Close
court
record
Tourney Soft
Teams
Filter
List view
Pool 1
Pool 2
Pool 3
Team
Team

Design Tradeoffs:

Problem: Consistency vs. flexibility among different sports.	**Solution:** A - Designing code for purely volleyball games & players B - Designing code that will support multiple sports	**Advantages:** A - Simpler initial developing and testing B - Multiple sports can be added with minimal changes to the code	**Disadvantages:** A - New classes will need to be added for other sports and the code must be built upon B - The initial development will be more complex and may take longer to complete	**Reasoning:** Solution A would be preferred because prioritizing a program that will effectively run tournaments for volleyball and will give us a stable core foundation before designing code for the flexibility of other sports
Problem: Designing a system that supports both in-depth stat collection and rapid score submission.	**Solution:** A - Prioritize the design for a detailed stat sheet (tracking every point, ace, kill) B - Prioritize a simple score entry and team winner	**Advantages:** A - Gives the users of the software a deeper view of the data B - Allows for a fast and easy way to maximize efficiency	**Disadvantages:** A - score inputs would be more time-consuming B - limited data means that users can't have an in depth game stat breakdown	**Reasoning:** Solution B would be preferred because it ensures that the program will run the most efficiently at the initial MVP
Problem: Creating a UI that supports a simple & easy to use interface while having extensive features	**Solution:** A - Keep the UI minimal and only feature essential features B - Implement a UI that is more feature- advanced for users such as spectators and fans	**Advantages:** A - Easier to navigate and easier to create B - Offers spectators a better user experience	**Disadvantages:** A - Only has the core features and doesn't include anything else B - Takes longer to design	**Reasoning:** Solution A would be preferred because it would be easier to design for the initial MVP