

Information Management & Systems Engineering

Milestone 1 – Specification

11718719 - Gruber Felix Benjamin
Wintersemester 2021

Inhalt

Domain Description	3
ER-Diagramm:	4
Use Cases:	5
Main Use Case 1 – Add game result	5
Main Use Case 2 – Player plays position(s).....	6
Use Case 3 – Player plays good with another player	7
Use Case 4 – Add player.....	8
Use Case 5 – Add Game	9
Reports.....	10
Report 1 – Best Players of a game:	10
Report 2 – Team Cadre:	10

Updates to Summer Semester 2021:

Since I attended the course last semester already, I thought I add a little section describing the (few) changes I did to implement the feedback from last semester (multiple seasons & report 1):

The whole system now only works for a singular season. Changing it so that it would work for multiple seasons would make the system too complex for this course. To achieve this, I:

- Changed the domain description.
- Removed the “season” Entity.
- Removed season from use case 5

New Report 1.

Domain Description

The goal of this project is to develop a small American Football database for the NFL (National Football League) universe. The system allows the user to keep track of the games and teams with their players and trainers of one season.

Each game has a start date time (which is the time of the kick-off) and the name of the stadium the game will take place. A home and a guest team will be assigned to each game. Last, but not least, it is possible to save the score of each team.

A football team consists of a team name, a division and a conference. There are two conferences in the NFL and four divisions per conference (these are somehow similar to the groups in a soccer World Cup).

Each team has different positions (e.g. "Quarterback" or "Receiver"). The position has a name, a short name (e.g. "QB" for "Quarterback") and is either an offensive, defensive or special team position. A position of a team can be played by one or multiple players.

Players are playing for football teams. Normally only for one team per season, but sometimes a player is traded to another team within a season. Each player may play on multiple positions of a team. The system can store the player's name, date of birth, height, weight, and (shirt) number. It is also possible to assign a score (1-100) to a player which indicates how good the player is.

Since some players are playing especially good together with other players it is possible to save this in the database. For each "plays good with"-connection between two players a score between (1 = worst and 5 = best) is assigned to indicate how strong the benefit of it is.

Last but not least, the system can store trainers. A trainer may be assigned to multiple teams, as it is the case for a player. It is possible to store the name, the date of birth, the job description (e.g. "Head Coach") and their start date in the NFL for a trainer.

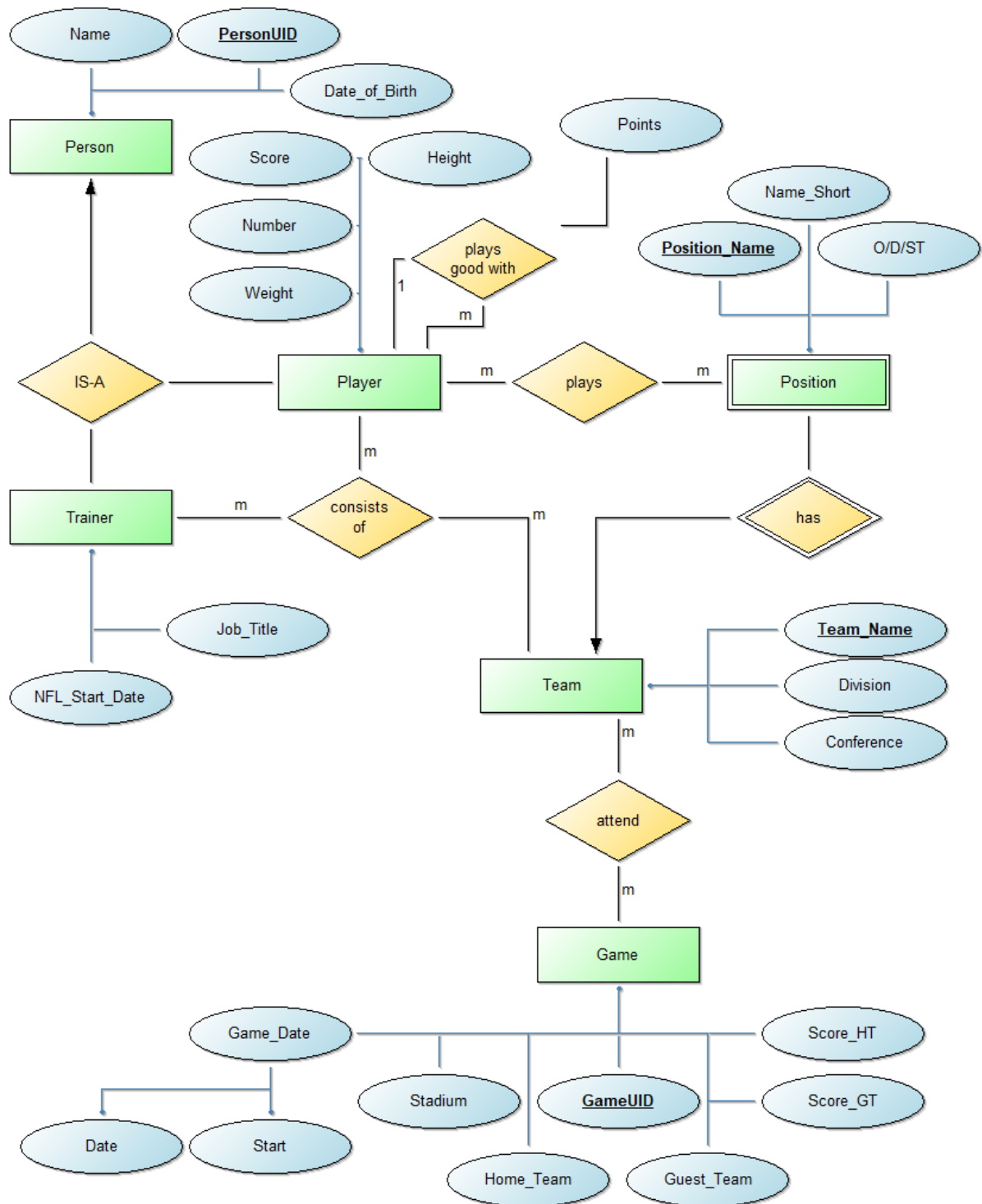
Use Cases:

- Main Use Case 1 – Add game result
- Main Use Case 2 – Player plays position(s)
- Use Case 3 – Player plays good with another player
- Use Case 4 – Add player
- Use Case 5 – Add Game

Reports:

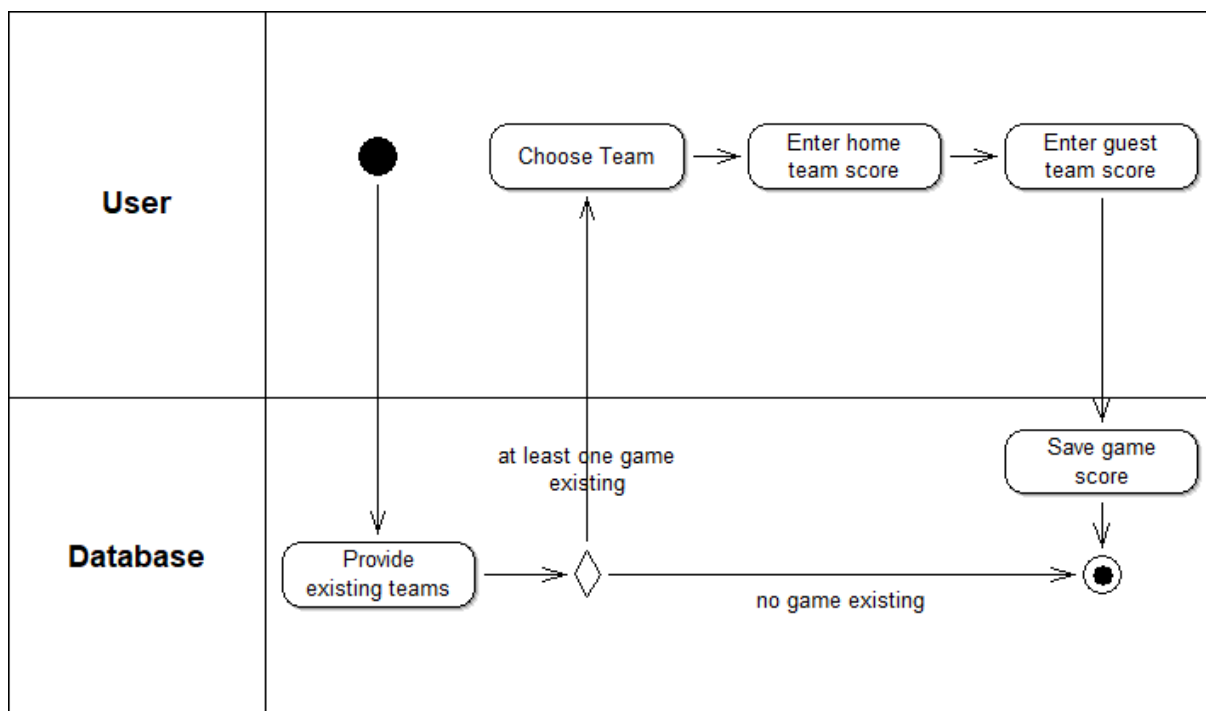
- Report 1 – Best Players of a game
- Report 2 – Team Cadre

ER-Diagramm:

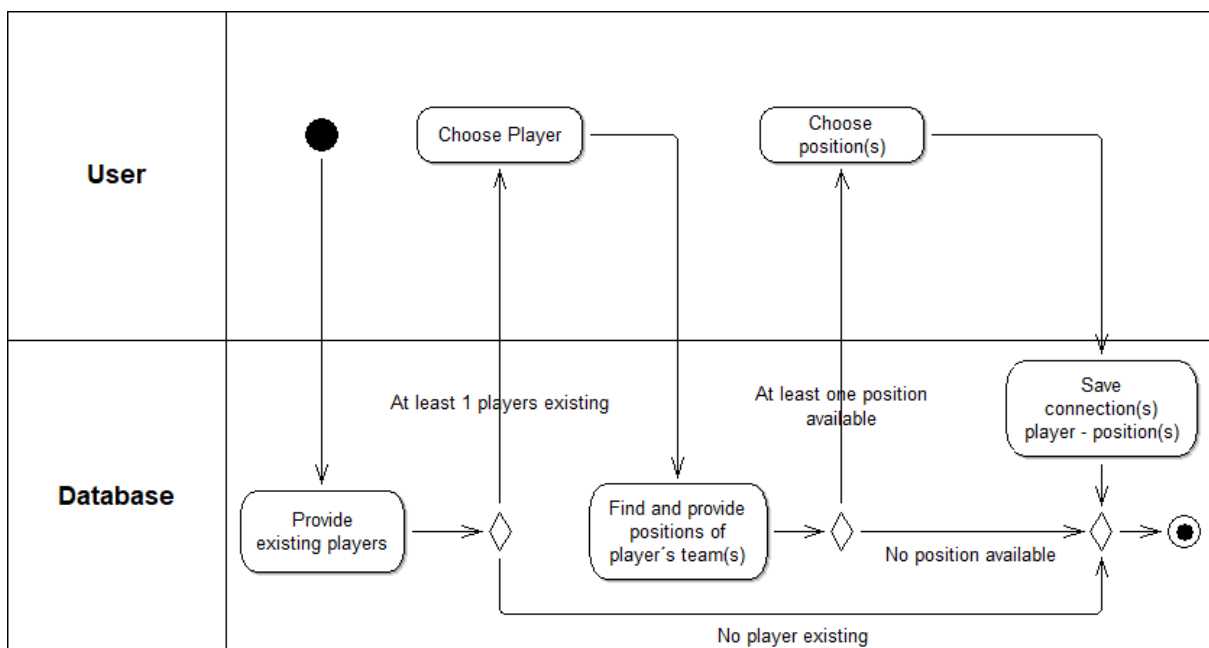


Use Cases:

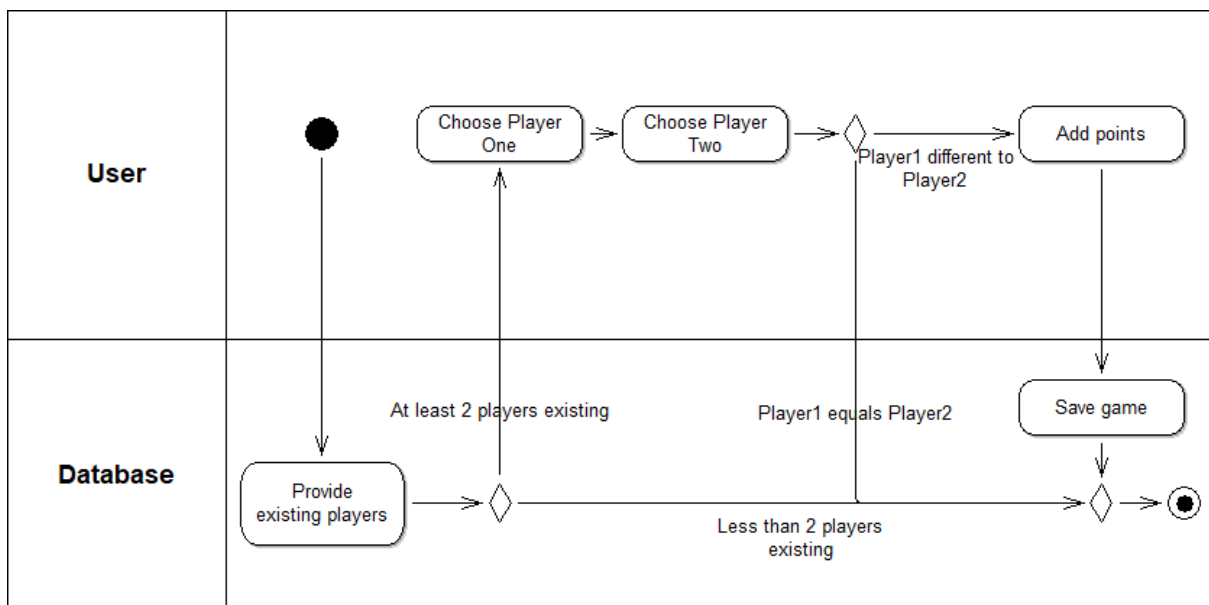
Main Use Case 1 – Add game result	
Objective	Both teams of the game have a score
Description	The games are entered as soon as the game plan is published. After a game took place, the scores of both teams should be added to the game. A score is always a positive integer (including 0).
Precondition(s)	<ul style="list-style-type: none"> - User is connected to database - Game is already present in the database
Expected Execution	<ol style="list-style-type: none"> 1. System provides existing games 2. User selects a game 3. User adds score for home team 4. User adds score for guest team 5. Data is stored in the database
Postcondition(s)	<p>Success:</p> <ul style="list-style-type: none"> - The score for both teams is saved to the game. <p>Error:</p> <ul style="list-style-type: none"> - The scores are illegal values: Scores are not added. Error message is shown to the user.



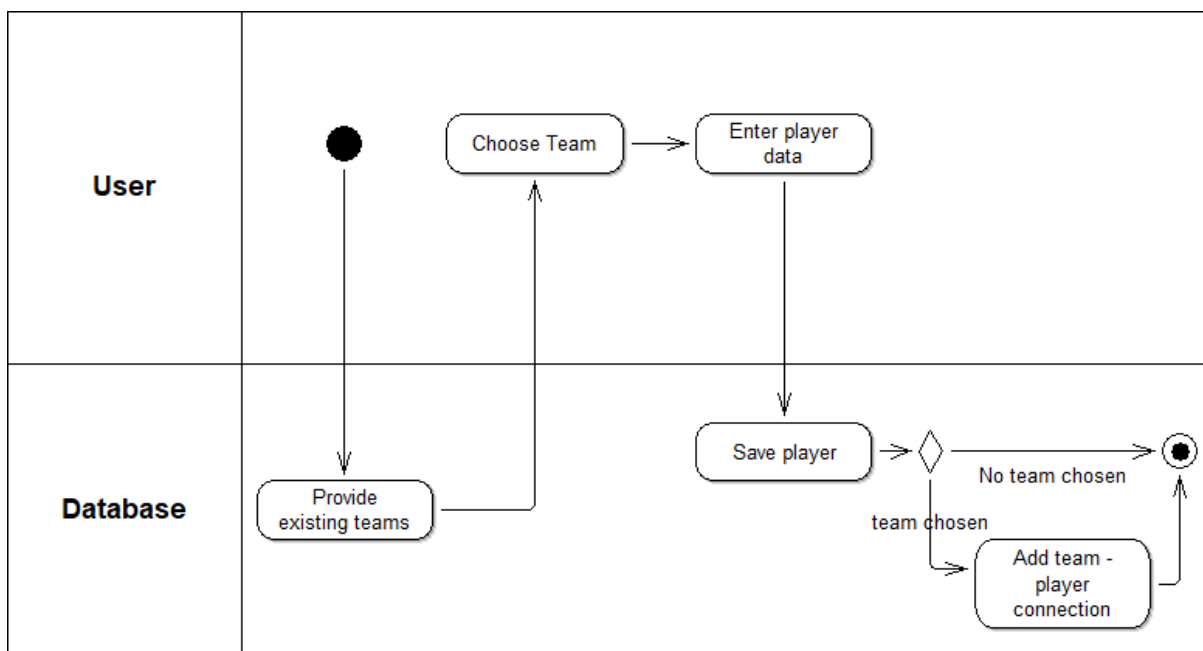
Main Use Case 2 – Player plays position(s)	
Objective	Connection between player and position is established
Description	The players and the positions are existing unrelated to one another. We now want to save the information that player X is playing on position Y.
Precondition(s)	<ul style="list-style-type: none"> - User is connected to database - Position(s) are already present in database - At least 1 Player is already present in database - Chosen Player is already part of a team
Expected Execution	<ol style="list-style-type: none"> 1. System provides available players 2. User chooses player 3. System provides positions of chosen player's team(s) 4. User chooses 1 or multiple positions of player's team(s) 5. Connection between player and positions is saved to system
Postcondition(s)	<p>Success:</p> <ul style="list-style-type: none"> - The connection between player and position(s) was successfully stored in the database. <p>Error:</p> <ul style="list-style-type: none"> - The new connection between player and position(s) was not saved. User is informed about error.



Use Case 3 – Player plays good with another player	
Objective	Add score for how good a player is playing together with another player
Description	If two players are playing especially good with one another. One can save this information to the database. The user has to select two users and adds a value that indicates how good they are playing together (5 best, 1 worst).
Precondition(s)	<ul style="list-style-type: none"> - User is connected to database - At least two players have to be present in the database
Expected Execution	<ol style="list-style-type: none"> 1. System provides available players 2. User selects 1. Player out of all existing players 3. User selects 2. Player out of all existing players 4. User selects value between 1 and 5 5. "Plays good with" connection is saved to database
Postcondition(s)	<p>Success:</p> <ul style="list-style-type: none"> - Connection between players and the associated points are saved in the database. <p>Error:</p> <ul style="list-style-type: none"> - 1. Player is same as 2. Player: No data is saved. Error message is shown to user. - Player(s) not available: No data is saved. Error message is shown to user. - Incorrect value for Points provided: No data is saved. Error message is shown to user.

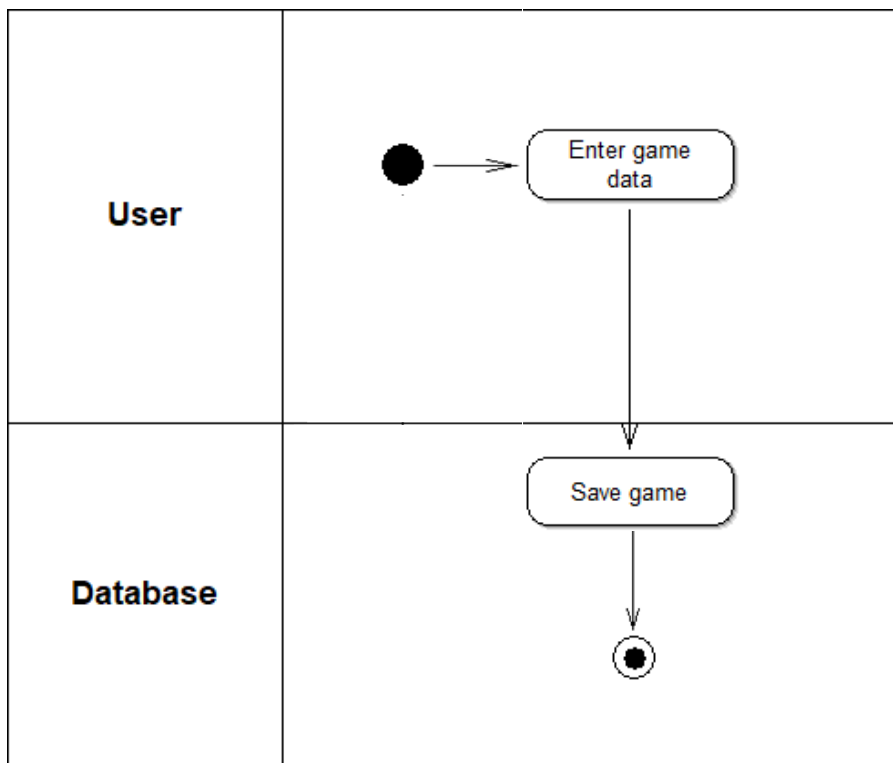


Use Case 4 – Add player	
Objective	New player is saved to the database
Description	A new player is added to the database. The user has to provide all player details. It should be possible to directly assign a player to one team, but it is not necessary. All player attributes are required.
Precondition(s)	<ul style="list-style-type: none"> - User is connected to database
Expected Execution	<ol style="list-style-type: none"> 1. System provides available teams 2. User may select a team 3. User provides necessary information for the player 4. Player is saved to the database 5. If team is given, connection between player and team is stored
Postcondition(s)	<p>Success:</p> <ul style="list-style-type: none"> - Player with all given information is saved to the database. <p>Error:</p> <ul style="list-style-type: none"> - Not all necessary details provided by user: Nothing is saved to the database. Error message is shown to the user. - Chosen team is not existing: Nothing is saved to the database. Error message is shown to the user.



Use Case 5 – Add Game

Objective	Add a new game to the database
Description	A new game is added. Normally a game is created before it took place. Therefore, the score is not available when the game is created. Every other attribute is required.
Precondition(s)	<ul style="list-style-type: none">- User is connected to database
Expected Execution	<ol style="list-style-type: none">1. User provides all necessary details for creating a game.2. Game is saved to the database
Postcondition(s)	<p>Success:</p> <ul style="list-style-type: none">- The game was added successfully to the database <p>Error:</p> <ul style="list-style-type: none">- Not all necessary game attributes provided: Game not added. Error message is shown to the user.



Reports

Report 1 – Best Players of a game:

Description:

This report should show the user the best players of a specific game. To achieve this, we get all players of both teams that attended the game. The players are filtered so that only players with a score of 4 or 5 are shown. The remaining players are sorted by their numbers.

Entities: Game, Team, Player

Field to be sorted: Player Number

Field to be filtered: Score

Report 2 – Team Cadre:

Description:

This report should show the user the players of a sub-team of a chosen team. The report is sorted by the position name so that it is easy to see which players are playing on a specific position. It is filtered for the sub-team type (offense/defense/special team).

Entities: Player, Position, Team

Field to be sorted: Position Name

Field to be filtered: O/D/ST