

TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Henry Veetamm IADB185486

Volleyball Club Management Web App

Distributed systems project

Supervisor: Andres Käver

Tallinn 2022

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Henry Veetamm

dd.mm.yyyy

Table of contents

Author's declaration of originality	2
Introduction	4
1. Overview	5
1.1. Application description.....	5
1.2. Web interfaces	6
2. Repository	10
3. ERD schema.....	11
Summary	12
References	Error! Bookmark not defined.

Introduction

I have been playing volleyball for 13 years. 12 years in youth classes and now on league.

Since I have played so much, I find I could have developed better skills when I was younger. After games I did not have any statistics about my performance in game and I did not get any feedback from workouts or volleyball matches either.

On one hand I see it is my coach's fault, but on the other hand I see that there is no proper and simple to use program/webpage for players and coaches on the same volleyball club

The aim of this project is to create an application where coaches and players can connect. Coaches can keep track of their players statistics and players can see how they are performing with some added functionality.

To fulfill this work, an ERD schema is created, then a database and then applications layers respectively to courses requirements.

1. Overview

1.1. Application description

This project is oriented to both of the coaches and players in a club.

Application makes coaches life easier to organize the club's overall life. For example, they can easily keep track of workout, add their matches performances etc. Coaches can easily manage their volleyball teams.

Players can see when they attended to workouts, their performance in volleyball matches. Also, they can compare themselves to others for extra competitive feelings.

The main goal is to make coaches life easier through this application. Now they can collect all their data into one place. I have seen that coaches still use the old-school way, using their notebook and pen. This was also me, because I have done it when I was U-20 boys volleyball team coach. I used Google Spreadsheet to keep track of who attended to workouts.

And the second goal is to let players know how they are doing overall. I have seen some teams who are keeping track of their players performances and players getting feedbacks which I believe greatly improves their themselves.

The idea is to give volleyball clubs a application that they can start using it out of the box and get better.

To finish this off, I should mention that this application is mainly for youth clubs, not for professional volleyball players, because every professional volleyball clubs have some sort of method to keep track their statistics and everything else.

1.2. Web interfaces

The main application will have two different views. One is for coaches and other for the players. If it is legally allowed to share the teams and clubs who have registered on the platform, then every person can see in the main screen all the registered clubs and teams. Concept is still very raw. (Figure 1)

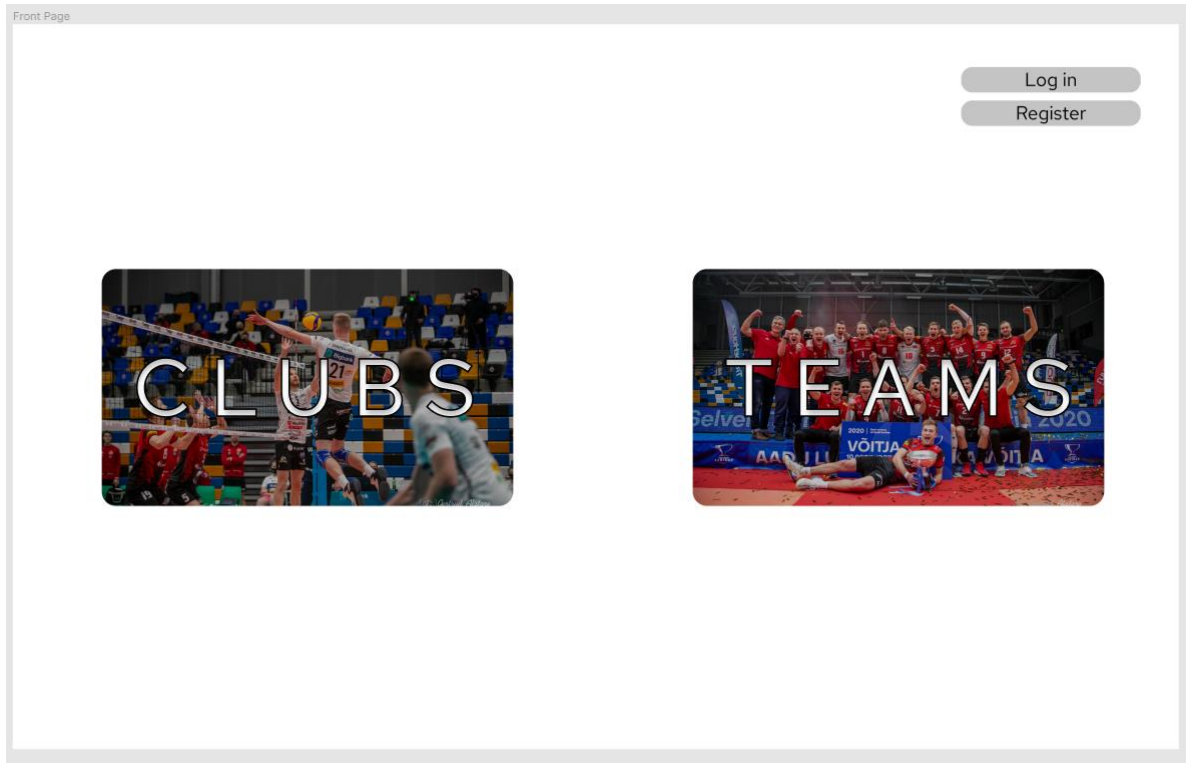


Figure 1. Home screen for everybody.

This is simple sketch for players. First idea is very simple, but I would love to create some universal way to calculate every players overall score, which is shown under “My matches” and “Saved players” views. (Figure 2)

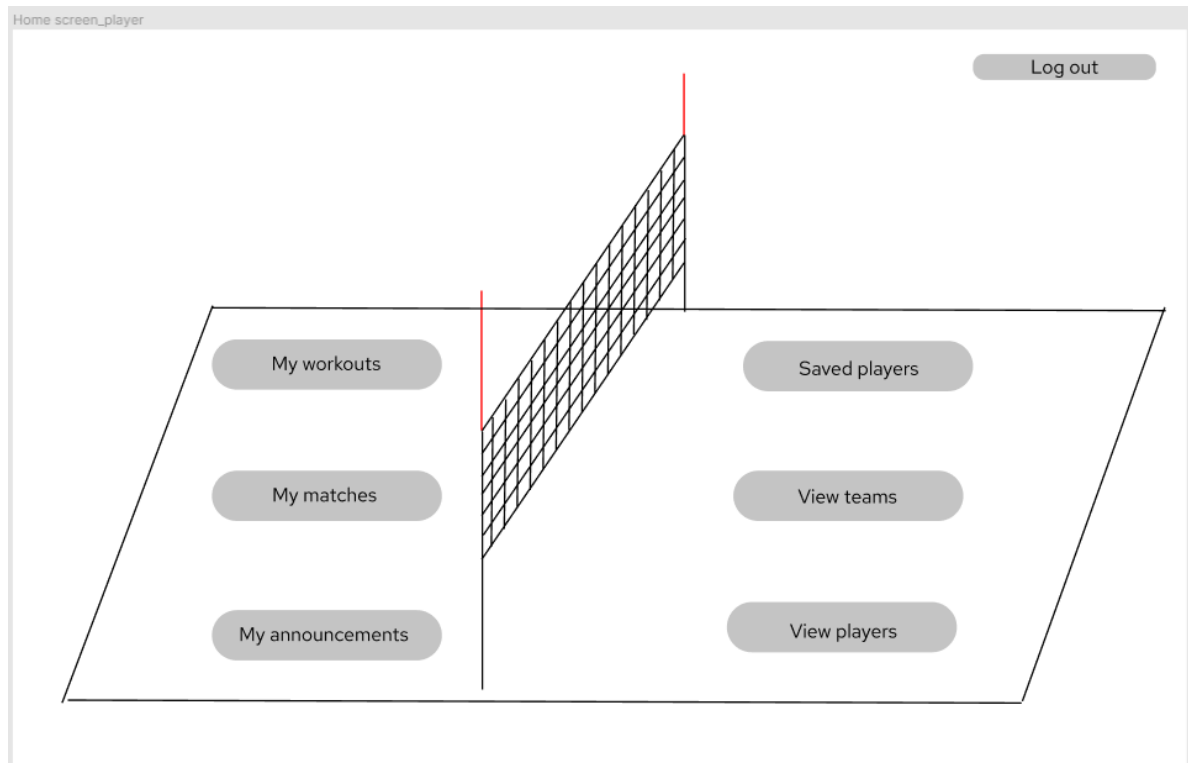
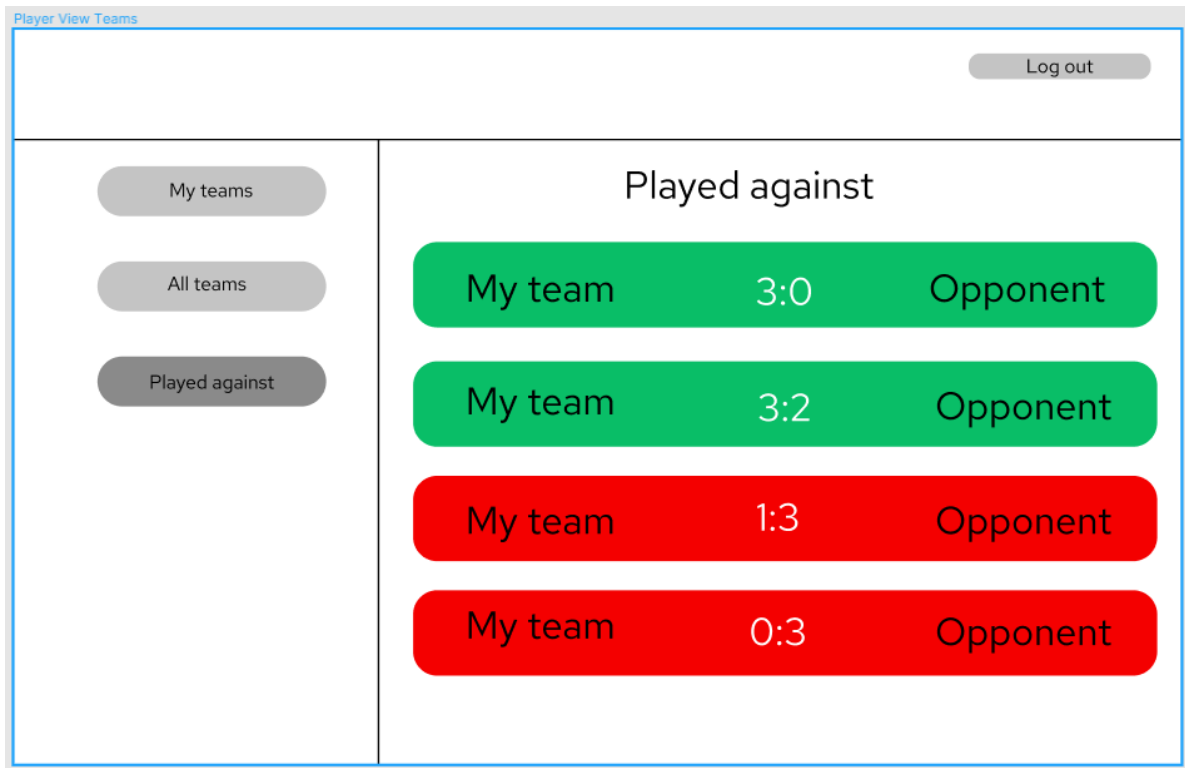


Figure 2. Players home view.

Figure 3. is showing “view teams” page, default selection would be “ My teams”. Currently on figure 3 it is showing “Played against” teams. Also when clicked on “My matches“ view from Figure 2, something similar is shown to user (“Played against” view)



This is coaches home screen where they can manage their everything about their clubs and teams. (Figure 3)

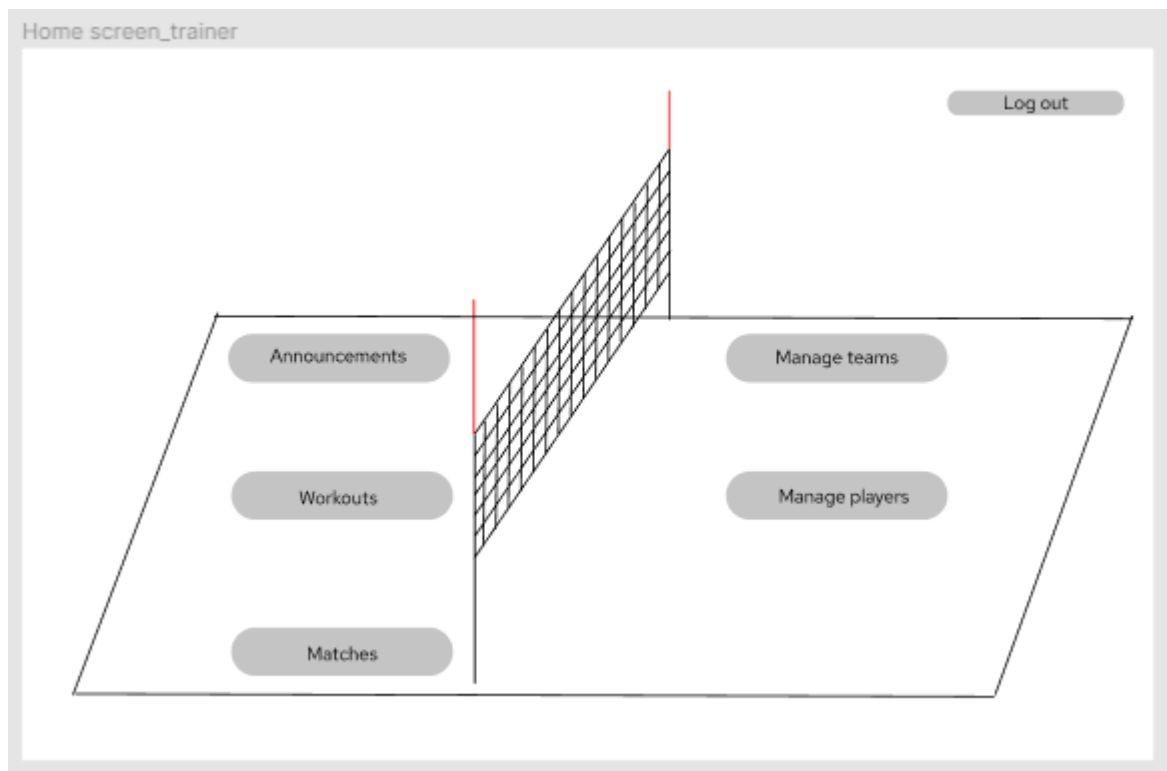
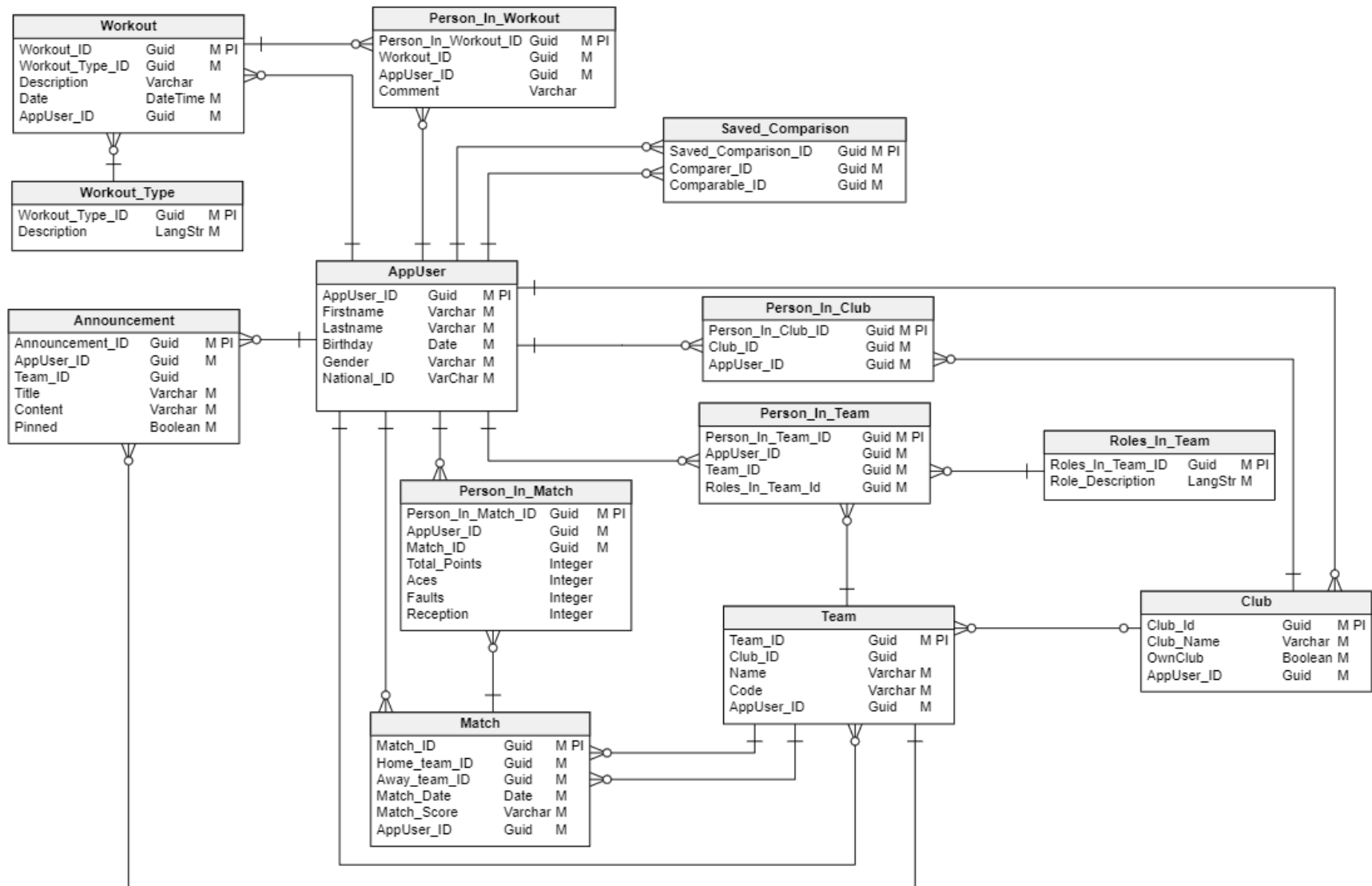


Figure 3. Coaches home screen.

2. Repository

<https://github.com/HenryVeetamm/Housing-app>

3. ERD schema



4. Table analysis

- PK – Primary key
- FK – Foreign key

4.1. Workout table

Only **coaches** can create and edit this table. If workout was created accidentally and no other players are related to this workout, only then a workout can be deleted.

Field	Nullable / Not Null	Explanation
Workout_ID, PK	Not Null	
Workout_Type_ID, FK	Not Null	
Description	Nullable	If something needs to be pointed out. For example if workout ended earlier. Or couch can describe what was done during the workout.
Date	Not Null	Indicates when workout was held.

4.2. Workout_Type table

Only admins can access this table. Admins could want to access this table if only other workout type is needed in the system. This table has predefined values. For example: cardio, gym/strength or ball drill. This table is needed if player wants to sort their workouts by types.

Field	Nullable / Not Null	Explanation
Workout_Type_ID, PK	Not Null	
Description	Not Null	

4.3. Person_In_Workout table

This table holds players workouts.

Field	Nullable / Not Null	Explanation
Person_In_Workout_ID, PK	Not Null	
Workout_ID, FK	Not Null	
AppUser_ID, FK	Not Null	
Comment	Nullable	Coach can note if player left the workout earlier, got injured or etc.

4.4. Announcement table

Coaches can view, create edit and delete the contents of this table. And players can view the announcements that are associated with them. This table can be used in many ways. For example to let **teams** know when a workout is dismissed, when next competition is held and etc. Or for example if coach wants to let know the whole club when the season ends or about different camps.

Field	Nullable / Not Null	Explanation
Announcement_ID, PK	Not Null	
AppUser_ID, FK	Not Null	Users ID (Coach), who created the announcement. If Team_ID is null then query to Person_In_Club is mandatory to determine to which club this announcement is made for.
Team_ID, FK	Nullable	Announcement can be made for a specific Team. If coach wants to make announcement to 2 teams, then some middle table is needed, but current schema doesn't allow it.
Title	Not Null	
Content	Not Null	
Pinned	Not Null	Pinned announcements appear first.

4.5. Saved_Comparison table

Players can save their comparisons. For example, Player A can compare themselves to Player B and save that for quicker access in future. Players can compare their matches, points and overall performance

Field	Nullable / Not Null	Explanation
Saved_Comparison_ID, PK	Not Null	
Comparer_ID, FK	Not Null	
Comparable_ID, FK	Not Null	

4.6. Person_In_Club table

Coaches can add registered players to volleyball club.

Field	Nullable / Not Null	Explanation
Person_In_Club_ID, PK	Not Null	
Club_ID, FK	Not Null	
AppUser_ID, FK	Not Null	

4.7. Person_In_Team table

Field	Nullable / Not Null	Explanation
Person_In_Team_ID, PK	Not Null	
AppUser_ID, FK	Not Null	
Team_ID, FK	Not Null	
Roles_In_Team, FK	Not Null	

4.8. Roles_In_Team table

Table with predefined roles. For example outside hitter, opposite. But if we are talking about younger players, they usually do not have any specific role, then we should need some neutral value or could make field „Roles_In_Team“ in „Person_In_Team“ table nullable.

Field	Nullable / Not Null	Explanation
Roles_In_Team_ID, PK	Not Null	
Role_Description	Not Null	

4.9. Team table

Coaches can manage (create, delete) their teams. If coach creates a team, then data is also inserted into „Person_In_Team“ table that shows that he is the coach of the team. A team, can be created without a club. Therefore „Club_ID“ can be a null. It's because if opponent team is not using our application then we can easily create a team so in future we can easily sort matches by team names. Other solution would be that in „Match Table“ (3.11) field „Away_Team_ID“ would be replaced with text, but if human error occurs on entering teams name then some results can lost when sorting

Field	Nullable / Not Null	Explanation
Team_ID, PK	Not Null	
Club_ID, FK	Nullable	
Name	Not Null	Teams name, for example VK Täht
Code	Not Null	Code field is used to identify team, U-16, U-18. If needed this field can be made into separate table with predefined values.

4.10. Club table

When coach registers his account then first thing to do is to register a club or join a club from that are already registered and after that coach can start to use the full application

Field	Nullable / Not Null	Explanation
Club_ID, PK	Not Null	
Club_Name	Not Null	

4.11. Match table

Only coaches can add matches and players can view only matches that they have participated.

Field	Nullable / Not Null	Explanation
Match_ID, PK	Not Null	
Home_Team_ID, FK	Not Null	
Away_Team_ID, FK	Not Null	
Match_Date	Not Null	
Match_Score	Not Null	This field shows the final score of a match with format („25:20,25:19, 25:18“, which mean home team won 3:0). Also could make two extra tables to track game score.(game, game_in_match)

4.12. Person_In_Match table

Field	Nullable / Not Null	Explanation
Person_In_Match_ID, PK	Not Null	
AppUser_ID, FK	Not Null	AppUser is player
Match_ID, FK	Not Null	
Total_Points	Nullable	All the points the player scored. Field is nullable because some coaches do not keep track of such data. And also some types of players can not score points (libero)
Aces	Nullable	Field is nullable because, some players do not serve (libero)
Faults	Nullable	
Reception	Nullable	

4.13. AppUser table

Provided by EF, added some extra mandatory fields.

Field	Nullable / Not Null	Explanation
AppUser_ID, PK	Not Null	
FirstName	Not Null	
LastName	Not Null	
Birthday	Not Null	
Gender	Not Null	
E-Mail	Not Null	
National_ID	Not Null	

5. Translation

Added support for static and database data translation. Created 12 resource files and translated doimain entities static fields.

Added custom datatype with name of LangStr, which supports translating user created values. LangStr is internally a dictionary which keys and values are strings and using PostgreSQL, which supports column type of jsonb. LangStr is saved as json to the database.

Example usage of LangStr:

```
await ctx.WorkoutTypes.AddAsync(entity: new WorkoutType()
{
    Description =
    {
        ["en"] = workoutTypeData.en,
        ["et"] = workoutTypeData.et
    }
}); // ValueTask<EntityEntry<...>>
```

Currently only WorkoutTypes and RolesInTeam tables are using a translation because as an end user, who is coach and player, are not going to enter the same data in different languages, that would be annoying for user. But as mentioned earlier, it is possible.