Group 9 Final Report

The demo is deployed on an Amazon EC2 instance (http://mocunye.com)

Group Members:

David Kerrigan(dwk55), Qi Cheng(qxc101), Jieyu Ren(jxr477), Mocun Ye(mxy293)

Name of the Application:

Esport Database

Example Application:

Liquipedia

1. Application Background:

There are many video games played competitively at the international level. Teams trade players and interact with other teams in tournaments and majors. Some players switch games. Teams have statistics such as the number of major wins, and players have statistics such as points per game. Our database will track and make simple tracking this complex scene for the user.

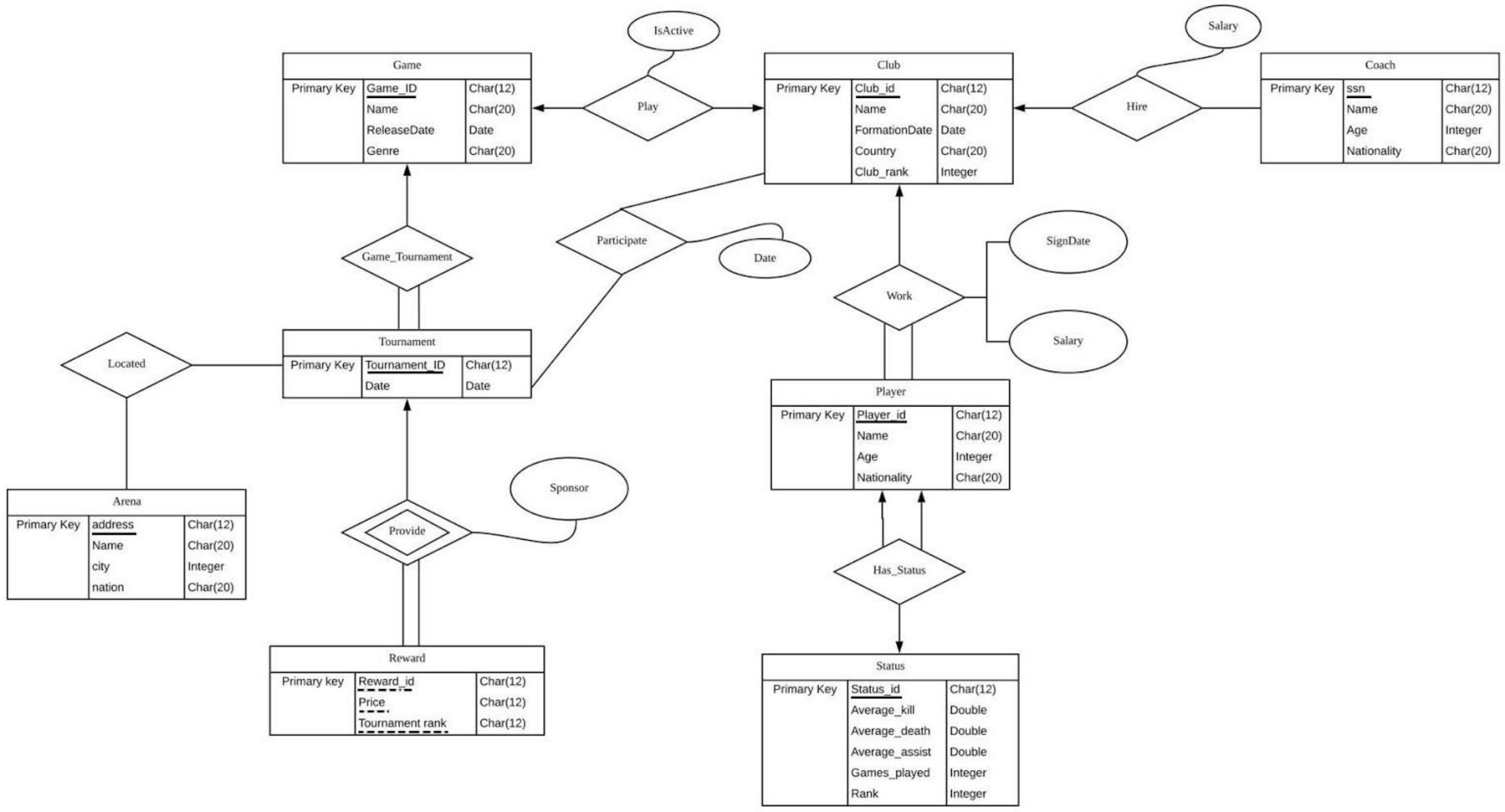
2. Data description

In this database, we'll be tracking the following entities: player, club, tournament, arena, reward, status, and coach.

The players will be signed onto a team and will have a status for overall statistics. Each player has basic attributes like name, age, nationality. Clubs will play certain games and hire coaches. Coach has similar attributes to player, and clubs have a formation date, whether they are active, nationality, and ranking.

Tournaments on specific dates will be held for specific games in an arena in a city and country, with a reward provided by a sponsor for a certain amount of money.

3. E-R Diagram



4. A list of functional dependencies derived from the semantics of your data,

Player:
{playerID, name, age, nationality }
playerID → name, age, nationality
Each playerID is correlated with one name, one age, and one nationality. PlayerID is the candidate key. BCNF form.

Status:
{statusID, average_kills, average_deaths, average_assists, games_played, rank}
statusID → average_kills, average_deaths, average_assists, games_played, rank
Each status is uniquely correlated with one average_kills, one average_deaths, one average_assists, one games_played and one rank. StatusID is the candidate key. BCNF form.

Has_Status:

{playerID, statusID}
playerID → statusID
Each player, uniquely identified by a player id, have only one set of batting statistics. PlayerID is the candidate key. BCNF form.

Games:
{gameID, name, releaseDate, genre}
gameID → name, releaseDate, genre
Each game, uniquely identified by a game id, have one name, one release date and one genre. GameID is the candidate key. BCNF form.

Play:
{clubID, gameID, teamFormationDate, isActive)
clubID → gameID, teamFormationDate,isActive
Each clubID is correlated with one game, one teamFormationDate, and one boolean called isActive since each club can only play one game, formed on a unique date, and can only be active or not. ClubID is the candidate key. BCNF form.

Clubs:
{clubID, name, formationDate, country, clubRank}
clubID → name, formationDate, country, clubRank
Each club, uniquely identified by a club id, have one name, one formation date and one country and one club rank. ClubID is the candidate key. BCNF form.

Work:
{playerID, clubID, signDate, salary}
playerID → clubID, signDate, salary
Each playerID is correlated with one clubID, one signed Date, and one salary because each tournament can only start on a unique date, end on a unique data, and be held at one place. playerID is the candidate key. BCNF form.

Tournament:
{TournamentID, StartDate, EndDate, Place}
TournamentID → startDate, endDate, place
Each TournamentID is correlated with one start date, one end date, and one place because each tournament can only start on a unique date, end on a unique data, and be held at one place. tournamentID is the candidate key. BCNF form.

Game_Tournament:
{TournamentID, GameID}
TournamentID → GameID
Each TournamentID is correlated with one gameID because each tournament plays a unique game. BCNF because tournamentID is a subset of the candidate key.

Reward_Provide:
{TournamentID, RewardID, Price, TournamentRank, Sponsor}
TournamentID, rewardID → Price, TournamentRank, Sponsor
Each TournamentID, rewardID is correlated with one Price, one tournament rank, and one sponsor because each tournamentID, rewardID has a unique price, unique tournament rank, and unique sponsor. BCNF because tournamentID, rewardID is the candidate key.

Participate:
{ClubID, TournamentID, Date}
clubID, tournamentID → date
Each clubid, tournamentId is correlated with one date. For a single clubid, tournamentid, there is a unique date of participation. BCNF because clubid, tournament is the candidate key.

5. The schema of your database, satisfying 3NF (if not BCNF)

Player:
{playerID, name, age, nationality }
PlayerID: id for the player entity set, primary key, not null.
Name: name of a player.
Age: age of a player.
Nationality: nationality of a player.

Status:
{statusID, average_kills, average_deaths, average_assists, games_played, rank}
statusID: id for the status entity set, primary key, not null.
Average_kills: average number of kills per game.
Average_deaths: average number of deaths per game.
Average_assissts: average number of assists per game.
Games_played: how many games played over career lifetime.
Rank: game-specific rank number, e.g. MMR for dota 2.

Has_Status:
{playerID, statusID}
PlayerID: id for the player set, primary key, foreign key references Player, not null.
StatusID: id for the status set, primary key, foreign key references Status, not null.

Games:
{gameID, name, releaseDate, genre}
gameID: id for the games entity set, primary key, not null.
Name: name of the game.
releaseDate: when the game was released.
Genre: genre of the game.

Play:
{clubID, gameID, teamFormationDate, isActive)
clubID: primary key for club entity set, primary key, foreign key references club, not null.
gameID: primary key for game entity set, primary key, foreign key references game, not null.
teamFormationDate: when this game division of the club was formed.
isActive: is this club division currently active?

Clubs:
{clubID, name, formationDate, country, clubRank}
ClubID: id for the club set, primary key, not null.
Name: name of a club.
FormationDate: the formation date of a club.
Country: the country that the club belongs to.
ClubRank: the rank of a club.

Work:
{playerID, clubID, signDate, salary}
playerID: primary key for player entity set, primary key, foreign key references player, not null.
ClubID: primary key for club entity set, primary key, foreign key references club, not null.
signDate: when the player signed with the club.
Salary: how much the club pays the player.

Tournament:
{TournamentID, StartDate, EndDate, Place}
TournamentID: the id for the tournament set, primary key, not null.
StartDate: the start date of the tournament.
EndDate: the end date of the tournament.
Place: the arena where the tournament is held.

Game_Tournament:
{TournamentID, GameID}
TournamentID: id for the tournament set, primary key, foreign key references Tournament, not null.
GameID: id for the game set, primary key, foreign key references Game, not null.

Reward_Provide:
{TournamentID, RewardID, Price, TournamentRank, Sponsor}
tournamentID: primary key for the tournament entity set, primary key, foreign key references tournament, not null.
rewardID: primary key for the reward entity set, primary key, foreign key references reward, not null
Price: the dollar amount of the reward

tournamentRank: major or minor tournament. How important is the tournament.
Sponsor: which company sponsors this reward?

Participate:
{ClubID, TournamentID, Date}
ClubID: id for club entity set, primary key, foreign key references Club, not null.
TournamentID: id for the tournament set, primary key, foreign key references Tournament, not null.
Date: the date that a club participate in a tournament.

6. Example queries supported by your system. You queries need to have some queries that use NOT EXISTS, EXCEPT as we practiced in the class

```
Select *
From Player

Select c.club_name From Club c
Where Not Exists ((Select p.club_id From Participate p
Where p.club_id = c.club_id)
Except
 (Select g.game_id From Game_tournament g
Where g.game_id = '000001'));

Select Player.player_name
From Player, Work, Club
Where Player.player_id =  Work.player_id
And Club.club_id = Work.club_id
And Club.name = “LGD”
And Player.nationality = “China”

Select Club.club_name
From Club, Play, Game
Where Club.club_id =  Play.club_id
And Game.game_id = Play.game_id
And Club.country = “US”
And Game.game_name = “Dota2”
```

7. Implementation

```
CREATE TABLE if not exist Arena
```

```
(
   arena_address character varying(20) NOT NULL,
   arena_name character varying(12),
   arena_city character varying(12),
   arena_nation character varying(20),
   CONSTRAINT arena_pkey PRIMARY KEY (arena_address)
)

CREATE TABLE if not exist Club
(
   club_id character varying(12) NOT NULL,
   club_name character varying(20),
   formation_data date,
   country character varying(20),
   club_rank integer,
   CONSTRAINT club_pkey PRIMARY KEY (club_id)
)

CREATE TABLE if not exist Coach
(
   coach_id character varying(12) NOT NULL,
   coach_name character varying(20),
   coach_age integer,
   nationality character varying(20),
   CONSTRAINT coach_pkey PRIMARY KEY (coach_id)
)

CREATE TABLE if not exist Game
(
   game_id character varying(12) NOT NULL,
   game_name character varying(20),
   release_data date,
   genre character varying(20),
   CONSTRAINT game_pkey PRIMARY KEY (game_id)
)

CREATE TABLE if not exist Game_tournament
(
   tournament_id character varying(12) NOT NULL,
   game_id character varying(12) NOT NULL,
   CONSTRAINT game_tournament_pkey PRIMARY KEY (tournament_id, game_id),
   CONSTRAINT game_tournament_game_id_fkey FOREIGN KEY (game_id)
       REFERENCES public.game (game_id),
   CONSTRAINT game_tournament_tournament_id_fkey FOREIGN KEY (tournament_id)
       REFERENCES public.tournament (tournament_id)
)

CREATE TABLE if not exist Has_status
(
   player_id character varying(12) NOT NULL,
   status_id character varying(12) NOT NULL,
   CONSTRAINT has_status_pkey PRIMARY KEY (player_id, status_id),
   CONSTRAINT has_status_player_id_fkey FOREIGN KEY (player_id)
       REFERENCES public.player (player_id),
   CONSTRAINT has_status_status_id_fkey FOREIGN KEY (status_id)
       REFERENCES public.status (status_id)
)

CREATE TABLE if not exist Hire
(
   club_id character varying(12) NOT NULL,
   coach_id character varying(12) NOT NULL,
   salary double precision,
   CONSTRAINT hire_pkey PRIMARY KEY (club_id, coach_id),
   CONSTRAINT hire_club_id_fkey FOREIGN KEY (club_id)
       REFERENCES public.club (club_id),
   CONSTRAINT hire_coach_id_fkey FOREIGN KEY (coach_id)
       REFERENCES public.coach (coach_id)
)

CREATE TABLE if not exist Is_located
(
   tournament_id character varying(12) NOT NULL,
   arena_address character varying(20) NOT NULL,
   CONSTRAINT is_located_pkey PRIMARY KEY (tournament_id, arena_address),
   CONSTRAINT is_located_arena_address_fkey FOREIGN KEY (arena_address)
       REFERENCES public.arena (arena_address),
   CONSTRAINT is_located_tournament_id_fkey FOREIGN KEY (tournament_id)
```

```
        REFERENCES public.tournament (tournament_id)
)

CREATE TABLE if not exist Participate
(
    club_id character varying(12) NOT NULL,
    tournament_id character varying(12) NOT NULL,
    participate_date date,
    CONSTRAINT participate_pkey PRIMARY KEY (club_id, tournament_id),
    CONSTRAINT participate_club_id_fkey FOREIGN KEY (club_id)
        REFERENCES public.club (club_id),
    CONSTRAINT participate_tournament_id_fkey FOREIGN KEY (tournament_id)
        REFERENCES public.tournament (tournament_id)
)

CREATE TABLE if not exist Play
(
    club_id character varying(12) NOT NULL,
    game_id character varying(12) NOT NULL,
    team_formation_date date,
    is_active boolean,
    CONSTRAINT play_pkey PRIMARY KEY (club_id, game_id),
    CONSTRAINT play_club_id_fkey FOREIGN KEY (club_id)
        REFERENCES public.club (club_id),
    CONSTRAINT play_game_id_fkey FOREIGN KEY (game_id)
        REFERENCES public.game (game_id)
)

CREATE TABLE if not exist Player
(
    player_id character varying(12) NOT NULL,
    player_name character varying(20),
    player_age integer,
    nationality character varying(20),
    CONSTRAINT player_pkey PRIMARY KEY (player_id)
)

CREATE TABLE if not exist Reward
(
    reward_id character varying(12) NOT NULL,
    price character varying(12),
    tournament_rank character varying(12),
    CONSTRAINT reward_pkey PRIMARY KEY (reward_id)
)

CREATE TABLE if not exist Reward_provide
(
    tournament_id character varying(12) NOT NULL,
    reward_id character varying(12) NOT NULL,
    sponsor character varying(12),
    CONSTRAINT reward_provide_pkey PRIMARY KEY (tournament_id, reward_id),
    CONSTRAINT reward_provide_reward_id_fkey FOREIGN KEY (reward_id)
        REFERENCES public.reward (reward_id),
    CONSTRAINT reward_provide_tournament_id_fkey FOREIGN KEY (tournament_id)
        REFERENCES public.tournament (tournament_id)
)

CREATE TABLE if not exist Status
(
    status_id character varying(12) NOT NULL,
    average_kills double precision,
    average_deaths double precision,
    average_assists double precision,
    games_played integer,
    status_rank integer,
    CONSTRAINT status_pkey PRIMARY KEY (status_id)
)

CREATE TABLE if not exist Tournament
(
    tournament_id character varying(12) NOT NULL,
    start_date date,
    end_date date,
    CONSTRAINT tournament_pkey PRIMARY KEY (tournament_id)
)

CREATE TABLE if not exist Work_for
(
    player_id character varying(12) NOT NULL,
    club_id character varying(12) NOT NULL,
    sign_date date,
    salary double precision,
    CONSTRAINT work_for_pkey PRIMARY KEY (player_id, club_id),
    CONSTRAINT work_for_club_id_fkey FOREIGN KEY (club_id)
        REFERENCES public.club (club_id),
    CONSTRAINT work_for_player_id_fkey FOREIGN KEY (player_id)
        REFERENCES public.player (player_id)
)
```

Insertion (only three examples for each table since there are too many of them):

```
insert into player values ('000001', 'Miracle-', 22,'Jordan');
insert into player values ('000002', 'w33', 24,'Romania');
insert into player values ('000003', 'MinD_ContRoL', 24,'Bulgaria');

insert into game_tournament values (000001, 000001)
insert into game_tournament values (000002, 000002)
insert into game_tournament values (000003, 000003)

insert into arena values ('Pudong Xinqu','Mercedes','Shanghai','China');
insert into arena values ('Banan','BLOOMAGE','Chongqing','China');
insert into arena values ('Harrison','Key','Seattle','US');

insert into is_located values ('000001','Pudong Xinqu');
insert into is_located values ('000002','Banan');
insert into is_located values ('000003','Harrison');

insert into reward_provide values ('000004','000004','Valve');
insert into reward_provide values ('000005','000005','Valve');
insert into reward_provide values ('000006','000006','Valve');

insert into reward values('000004','1763200','major');
insert into reward values('000005','198442','minor');
insert into reward values('000006','2364200','major');

insert into participate values ('000001','000001','2019-01-26');
insert into participate values ('000002','000001','2019-01-26');
insert into participate values ('000003','000001','2019-01-26');

insert into Plays values ('000001','000001','10-22-2010', True);
insert into Plays values ('000001','000003','02-03-2009', True);
insert into Plays values ('000002','000002','01-06-2015', True);

insert into Clubs values ('000001', 'Natus Vincere', '01-01-2009', 'Ukraine', '1');
insert into Clubs values ('000002', 'Team Liquid', '01-06-2000', 'US', '2');
insert into Clubs values ('000003', 'Evil Geniuses', '02-02-1999', 'US', '3');

insert into tournament values ('000003', '2020-10-5','2020-10-12');
insert into tournament values ('000003', '2020-10-5','2020-10-12');
insert into tournament values ('000004', '2020-10-5','2020-10-12');

insert into coach values ("000001", "Pipat Prariyachat", 30, "Thailand"');
insert into coach values ("000002", "William Lee", 29, "US")
insert into coach values ("000003", "Andrey Chipenko", 29, "Ukraine")

insert into game values ("000001", "Dota2", "2013-09-07", "US")
insert into game values ("000002", "League Of Legends", "2009-10-27", "US")
insert into game values ("000003", "CSGO","2012-08-21", "US")

insert into has_status values ('000001',1);
insert into has_status values ('000002',2);
insert into has_status values ('000003',3);

insert into status values
(1,12.871248896536592,9.830679948176913,5.541685784782785,11850,6628);
insert into status values
(2,12.912346195108915,11.204006522036986,1.4142747899004204,9576,7029);
insert into status values
(3,14.595179713580945,10.038919489583165,4.364032416565358,8335,7029);

insert into hire values ("000001", "000001", 62000)
insert into hire values ("000002", "000003", 68000)
insert into hire values ("000003", "000002", 63500)

insert into work_for values ("000001", "000002", "2010-01-06", 70000)
insert into work_for values ("000002", "000002", "2010-01-06", 70000)
insert into work_for values ("000003", "000002", "2010-01-06", 70000)
```

8. The role and contributions of each team member

David Kerrigan:
Data insertion, helped design schema and E-R diagram. Worked on presentation and reports.

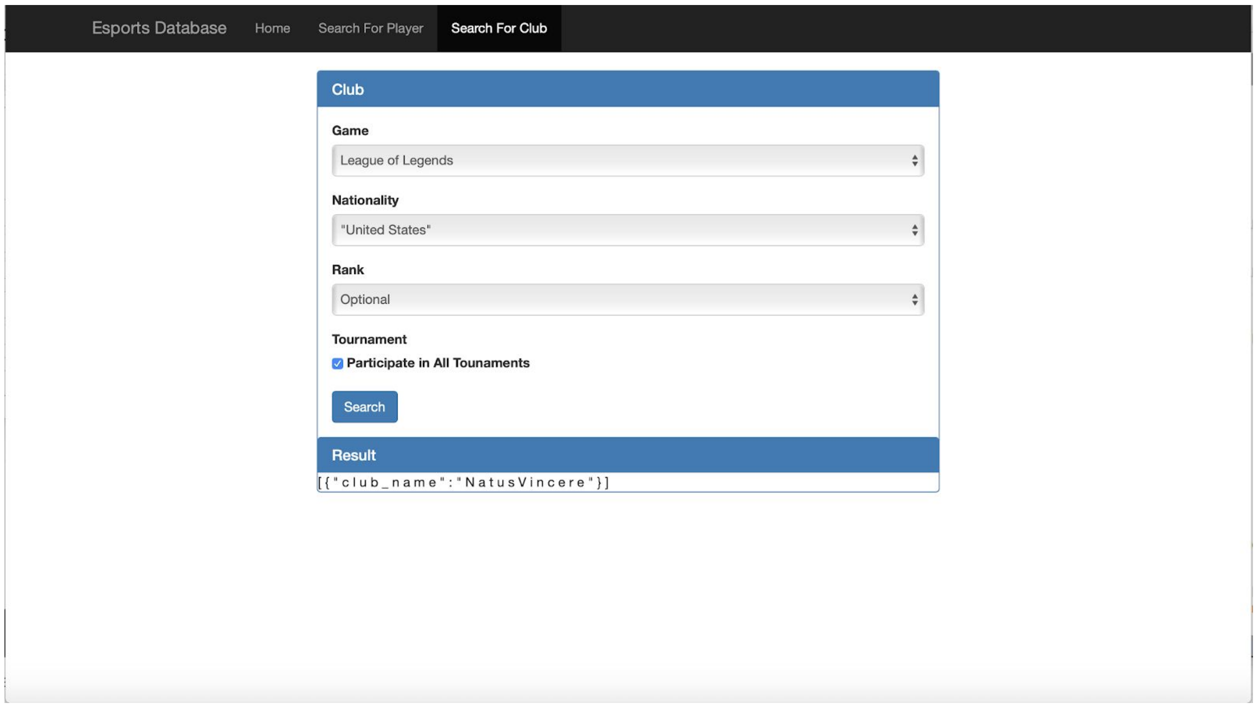
Mocun Ye:
Writing front-end and back-end web UI and the Python functions to generate the queries. Helped to design the schema and the E-R diagram, worked on presentation and reports.

Jieyu Ren:
Constructed the database, manually inserted data, helped to design the schema and the E-R diagram, worked on presentation and reports.

Qi Cheng:
Data insertion, schema design, E-R diagram design, worked on presentation and reports.

9. What you have learned from this project — over and above what is covered in the course.
This project was pretty fun, and it was our first time working with nosql, pgadmin, and flask. We feel that I learned useful software tools and skills during this project. We learned how to create a postgresql database with constraints and keys, as well as analyze the FD and normal forms of the entity and relationship sets. We learned how to query the postgresql database using python and psycopg2 and how to create an intuitive web UI with python and flask which queries and retrieves responses from a database.

Appendix of Screenshots



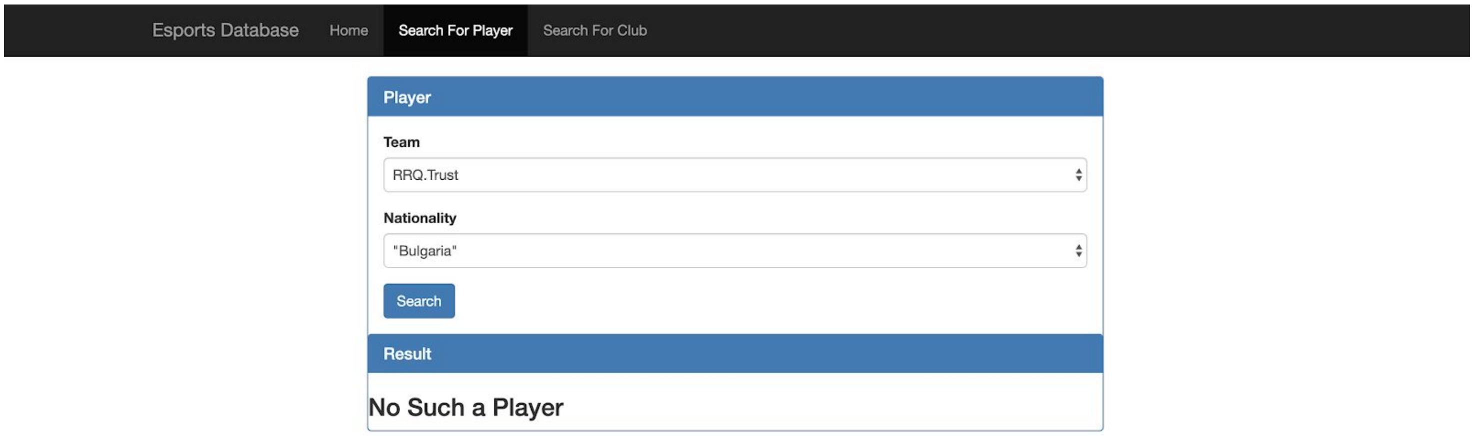
Game: League of Legends

Nationality: "United States"

Rank: Optional

Tournament: ✓ Participate in All Tounaments

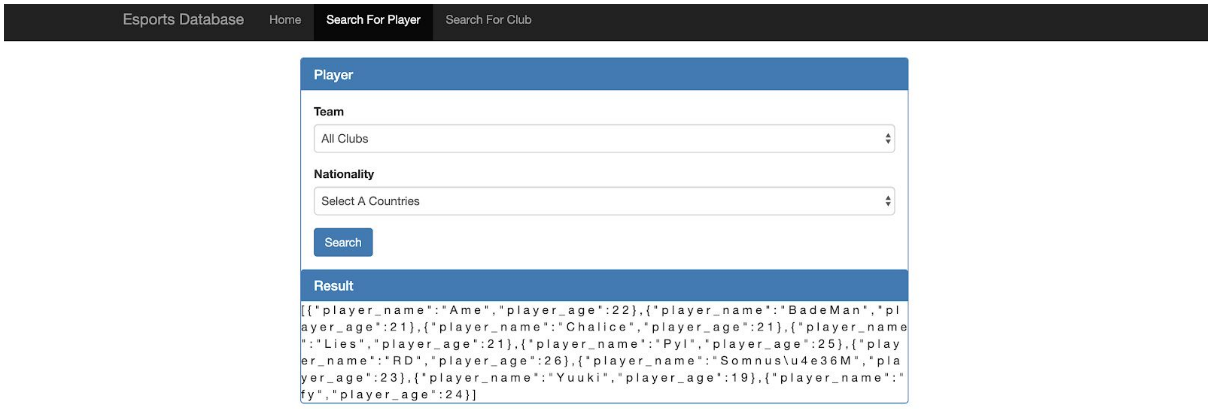
```
[{"club_name":"NatusVincere"}]
```



Team: RRQ.Trust

Nationality: "Bulgaria"

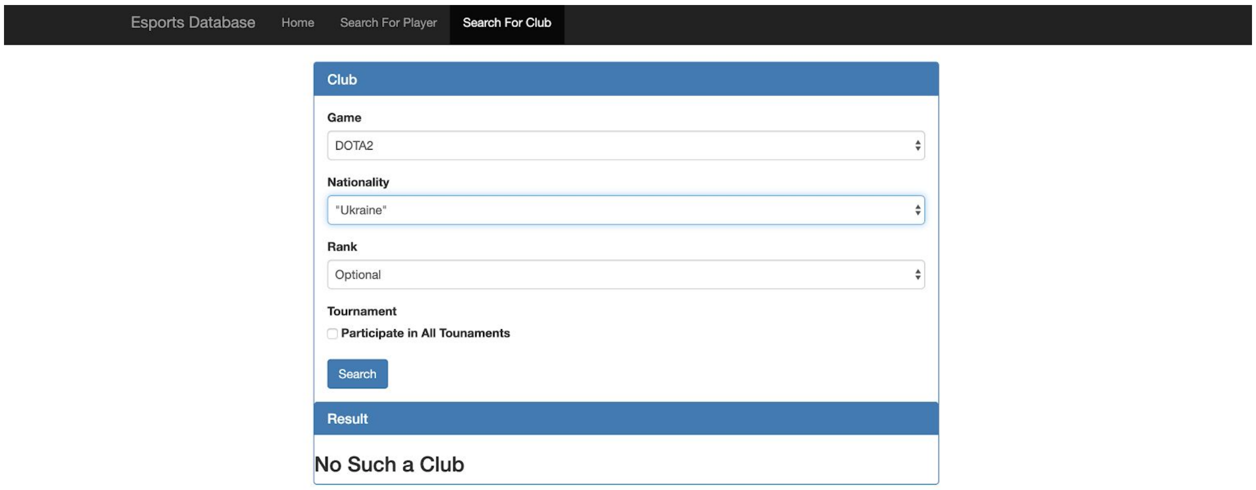
No Such a Player



Team: All Clubs

Nationality: Select A Countries

```
[{"player_name":"Ame","player_age":22},{"player_name":"BadeMan","pl
ayer_age":21},{"player_name":"Chalice","player_age":21},{"player_name
":"Lies","player_age":21},{"player_name":"Pyl","player_age":25},{"play
er_name":"RD","player_age":26},{"player_name":"Somnus\u4e36M","pla
yer_age":23},{"player_name":"Yuuki","player_age":19},{"player_name":"
fy","player_age":24}]
```



Game: DOTA2

Nationality: "Ukraine"

Rank: Optional

Tournament: Participate in All Tounaments

No Such a Club