

1. Cover Page

University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Part 3

Date: Nov 20, 2020

Project Group Number on Canvas: 101

Group Members:

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Vandy Liu	43550870	x8s1b	vandyliu1999@gmail.com
Eric Jang	54876453	n1l1b	ericjang310@gmail.com
Wilson Gee	69247682	y4m1b	y4m1b@ugrad.cs.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Codebase

Attached to the Canvas submission is our zipped project.

Also retrievable at <https://github.com/vandyliu/304project>

3. SQL Script

In the /sql_scripts folder.

Found at https://github.com/vandyliu/304project/tree/master/sql_scripts

4. Project Description

The domain of our application is in the world of online video games, particularly the new competitive video game: Valorant. It is for Valorant players, fans, tournament organizers, fanatics in general to easily find more information about their matches, and their account as well as other people's accounts and matches in a single nice website.

Our application will keep track of players, the matches they play in and tournaments they join. This would include a player's career statistics (e.g. a player would be able to see how they have performed over the past few weeks), and also information on the esports scene, so fanatics and analysts are able to follow tournaments more closely. We would also be keeping track of cosmetics they purchase or earn. Tournament organizers could easily display match information on streams and events. It can also help track a player's progress as they play over time to see how they improved. Newer players can find information on the playable characters, also known as agents, on our website to get a feel of which ones they might want to try out when they're just starting.

Our service can be incorporated into Valorant's servers or as a standalone service. With 3 million unique players logging in to play daily, the abundance of matches every hour, and the constant updates to data like when a game is played, when a weapon skin is bought, a database would provide the performance needed to ensure that the service can be sustained. Our application would be constantly updated with tons of requests, and a DBMS is the only sensible way to provide these functionalities. Because many gamers are competitive and like to keep track of their stats, our service can feed a player's competitive spirit by comparing their stats with friends as well as keep track of their progression as they become a more skilled player.

In our final product, we had to add an extra field 'average_combat_score' to the 'Player' table. Initially, we did not include it as we argued that it could simply be calculated using existing fields in our 'Player' schema, namely using the fields: kills, assists, damage_dealt, and total rounds played. However, we realized that it would be far more efficient to avoid doing a calculation every time someone wanted to view their average_combat_score, especially since this would require a join on multiple tables. Because a player's strength cannot simply be assessed by the

means of a kill-death ratio as they could be playing selfishly in favour of statistics instead of for the team, players may feel more inclined to turn towards the average_combat_score to more accurately assess their competency. Ultimately, this additional field allows us to provide users of the application with greater insight and simultaneously less overhead

Pages

- **/**
 - home page
- **/Agents**
 - shows a list of the available agents in the game
- **/Tournaments**
 - shows a list of all the tournaments that have happened
 - also shows list of organizers for the tournaments (Aggregation - Group By), their first tournament date, and the total amount of money they put into the prize pools for their tournaments
- **/Team**
 - shows list of teams in the database
 - can create a team here to add (Insert)
 - can also update a team here to change their wins, losses or name (Update)
 - can also filter to see which teams have played in which tournaments, or teams that have played in all tournaments (Division)
 - can also see each team's tournament history by clicking on the tournament history button (goes to /Team/:team_id)
- **/Team/:team_id**
 - view the team's tournament history (Join: Team, Team_Tournament)
- **/Players**
 - shows a list of players in the database
 - can select which columns you want to show by clicking a button (Projection)
 - can also filter players by rank, minimum kills, assists, death, average combat score (ACS), headshot percentage or a combination of them (Selection)
 - can also delete a player from the database here (Delete)
 - can view a player's match history (goes to /Players/:player_id)
 - there's also a button at the bottom where you can see the average ACS for each rank, where each rank's average ACS must be greater than the average ACS across all ranks (Aggregation - Nested)
- **/Players/player_id**
 - view match history for the player (Join: Match_Player, Player table)
 - can filter by map, gamemode or agent
 - can see average stats per map, gamemode or agent (Aggregation - Having)
 - only where count for the group is greater than 3 to rule out anomalies
- **/Matches/**
 - view all the matches

5. SQL Queries used

“Insertion: add a player to the players list”

“Deletion: remove an ability from an agent”

“Update: change the experience gained from a mission”

“Selection: select all players with radiant rank and have more deaths than kills”

“Projection: find ranks, kills, and ID of all players”

“Join: look at (player) BigEdKnorr’s stats when he plays matches as the agent, Reyna on Bind”

“Division: Find ids of all matches in a tournament organized by RiotGames”

Aggregation with Group By

– Create one query that requires the use of distinct aggregation (min, max, average, or count are all fine), and provide an interface (e.g., HTML button/dropdown, etc.) for the user to execute this query.

Aggregation with Having – Create one meaningful query that requires the use of a HAVING clause, and provide an interface (e.g., HTML button/dropdown, etc.) for the user to execute this query.

Nested Aggregation with Group By

Queries to have in our milestone 3

- Insertion: Create new team with a button and form
 - Eg. Create new team G2 with 0 wins, 10 losses
- Deletion: Remove player by pressing delete button near player (eg. they were banned for cheating or quit the game)
 - It is an ON DELETE CASCADE: you will notice the team has one less player
 - Eg. Remove BigEdKnorr for hacking
- Update: You can update a team’s name, wins, losses, etc. on the teams page
 - Eg. Change Cloud9’s team name to Cloud9 Blue
- Selection: Select all players with a specific rank, or with at least X kills, deaths, assists, headshot percentage
 - Eg. Show me all the players that have at least 10 kills and are in Bronze rank
- Projection: Choose what columns you want for the players table

- Eg. Show me the kills and deaths of all players
- Join: Matches are joined with match_players so we can see match history of a player
 - Eg. look at (player) BigEdKnorr's stats when he plays matches as the agent, Reyna on the map, Bind
- Division: Find ids of all matches in a tournament organized by RiotGames
- Aggregation with Group By: Find aggregated details of organizers that hosted a tournament
 - Eg. Find total prize pool of all tournaments for each organizer
- Aggregation with Having:
 - Eg. Find avg kills, avg deaths, etc grouped by map for a player, BigEdKnorr
 - Is good for knowing whether a player plays better on specific maps, agents, etc
- Nested Aggregation with Group By: ??????????
- get avg avg combat score for each rank where the avg avg combat score is higher than the avg avg combat score across all ranks

6. Sample [Screenshots](#)

All photos are in the /screenshots folder. There is also a GIF there.

There is also a gif in the README and screenshots folder that shows pretty much all features.

SELECT COLUMNS

☒ Rank ☒ Kills ☒ Assists ☒ Deaths ☐ Headshot % ☒ AverageCombatScore

PLAYERS

player_id	p_rank	kills	assists	deaths	average_combat_score		
C9 Mitch#000	Platinum 3	543	132	131	218	MATCH HISTORY	DELETE
C9 Relys#000	Platinum 2	76	23	13	212	MATCH HISTORY	DELETE

SHOW NESTED AGGREGATION

AVERAGE ACS PER RANK

p_rank	AvgACS
Diamond 2	203.3333
Platinum 3	218
Platinum 2	212
Radiant	221.4286
Immortal 3	209.5
Immortal 1	208
Immortal 2	230



FIND PLAYERS

Rank: All Ranks Min. Kills: Min. Assists: 1 Min. Deaths: Min. Headshot %: Min. ACS:

SELECT COLUMNS

☒ Rank ☒ Kills ☒ Assists ☒ Deaths ☐ Headshot % ☒ AverageCombatScore

PLAYERS

player_id	p_rank	kills	assists	deaths	average_combat_score		
BigEdKnorr#572	Diamond 2	2015	185	973	201	MATCH HISTORY	DELETE
C9 Mitch#000	Platinum 3	543	132	131	218	MATCH HISTORY	DELETE
C9 Relys#000	Platinum 2	76	23	13	212	MATCH HISTORY	DELETE
C9 Shinobi#000	Radiant	131	56	178	154	MATCH HISTORY	DELETE
C9 TenZ#000	Radiant	9000	4555	2455	299	MATCH HISTORY	DELETE
C9 Vice#000	Radiant	566	655	544	199	MATCH HISTORY	DELETE
gamer#123	Iron 1	12	3	46	85	MATCH HISTORY	DELETE

FILTER TEAMS

Tournament

Any Tournament ▾

TEAMS

G2	Wins: 1 Losses: 0	⌵
<div>EDIT TEAM</div> <div>TOURNAMENT HISTORY</div> <div>BigEdKnorr#572 gamer#123 SmallEdKnorr#572 TenZ#000</div>		
TSM	Wins: 12 Losses: 2	⌵
Sentinels	Wins: 15 Losses: 1	⌵
Team Envy	Wins: 5 Losses: 1	⌵
Cloud9	Wins: 400 Losses: 6	⌵
CPSC304	Wins: 100 Losses: 1	⌵
<div>CREATE NEW TEAM</div>		

FILTER TEAMS

Tournament

Any Tournament ▾

TEAMS

G2		⌵
<div>Update Team</div> <div>Enter the team details below.</div> <div>Team ID 1</div> <div>Name G2</div> <div>Wins 1</div> <div>Losses 0</div> <div>CANCEL UPDATE</div>		
TSM	Wins: 5 Losses: 1	⌵
Sentinels	Wins: 400 Losses: 6	⌵
Team Envy	Wins: 100 Losses: 1	⌵
Cloud9		⌵
CPSC304		⌵
<div>CREATE NEW TEAM</div>		

FILTER TEAMS

Tournament

Any Tournament ▾

TEAMS

G2	Wins: 1 Losses: 0	⌵
EDIT TEAM		
TOURNAMENT HISTORY		
BigEdKnorr#572 gamer#123 SmallEdKnorr#572 TenZ#000		
TSM	Wins: 12 Losses: 2	⌵
Sentinels	Wins: 15 Losses: 1	⌵
Team Envy	Wins: 5 Losses: 1	⌵
Cloud9	Wins: 400 Losses: 6	⌵
CPSC304	Wins: 100 Losses: 1	⌵
CREATE NEW TEAM		

CPSC304 TEAM 101