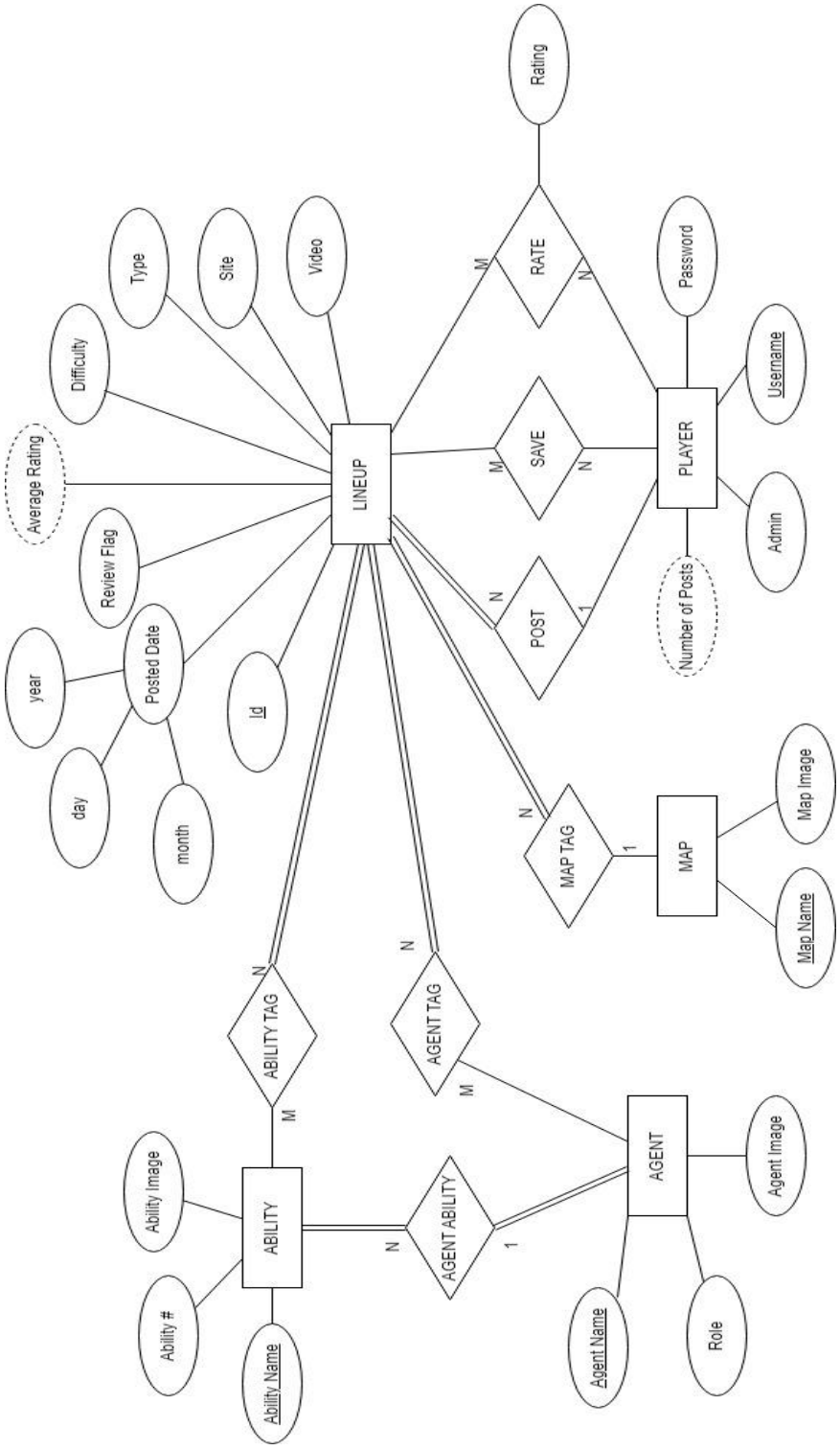
Valorant Lineup Database

Ayman Rahman

Problem Statement:

There is currently no centralized platform or database for Valorant lineup videos that includes comprehensive annotations and instructions for each lineup. Players who want to learn lineups for different agents and maps have to search through various social media platforms and video sharing websites, which can be time-consuming and inefficient. Additionally, many of these videos may not have enough information or may not be up-to-date. The solution is to create a database specifically for Valorant lineup videos that includes comprehensive annotations and instructions for each lineup. The database should have a review system where players can rate and provide feedback on the usefulness and accuracy of each video. It should also have a tagging system where players can search for videos based on different criteria such as agent, map, skill level, and type of lineup. By creating a centralized and comprehensive database for Valorant lineup videos, players can improve their gameplay and enjoy the game even more.

Conceptual Database Design:



The ‘Player’ entity is the collection of players who have registered an account. They are able to post their own lineups, save other player lineups, and rate other players lineups.

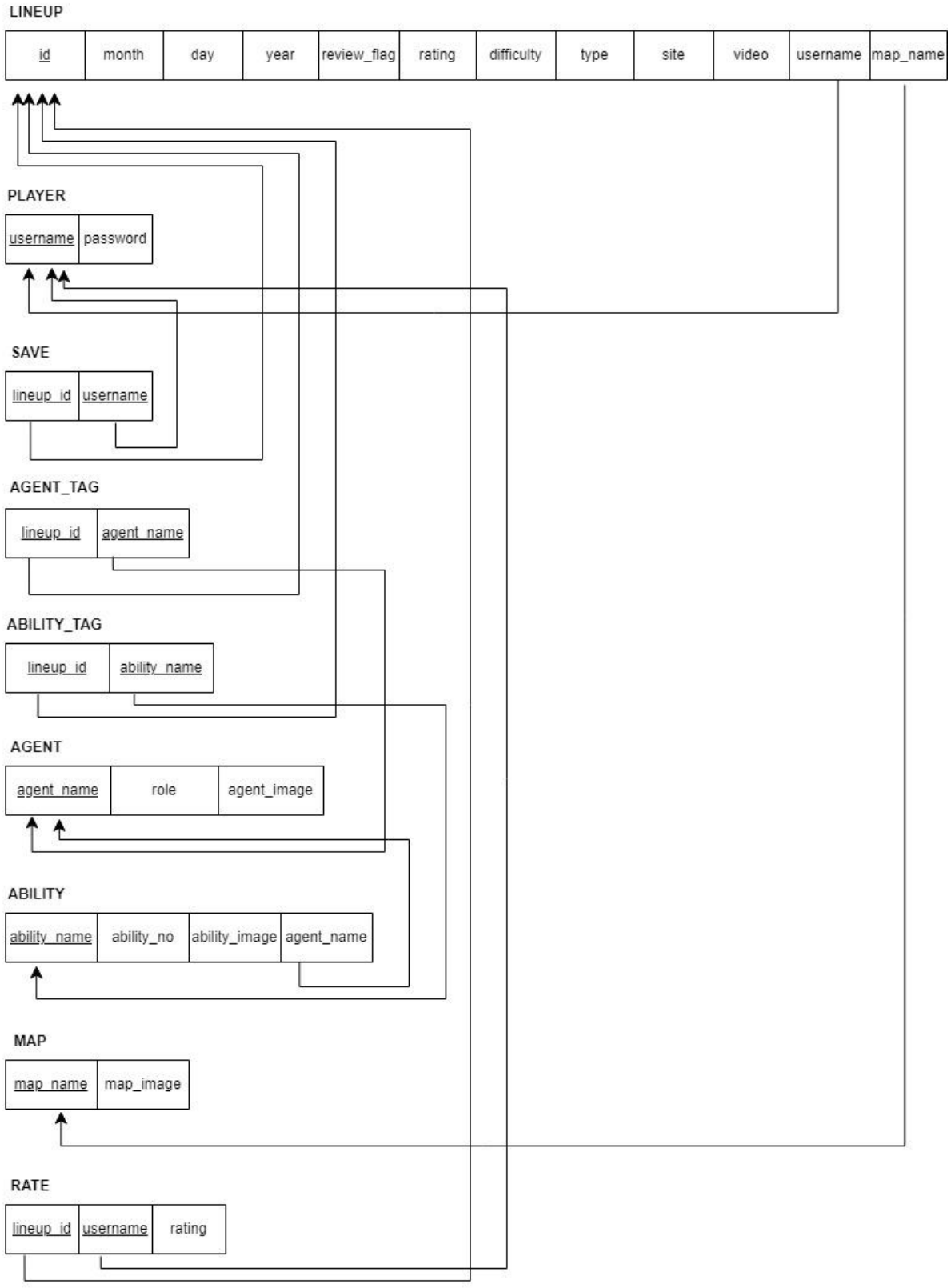
The ‘Lineup’ entity stores the video links and information tied to the video. A lineup can be tagged with a map, agent, and abilities. A lineup can also be flagged for review so an admin can review it to deem it as misinformation.

The ‘agent’ entity has information tied to it and an image that can be used to display its icon
An agent has multiple abilities tied to them.

The ‘Ability’ entity has information tied to it as well as an image that can bemused to display its icon

The ‘Map’ entity is used to be tagged under lineups and store its image so that it can be displayed for a lineup.

Logical Database Design:



Summary Table of Data Types:

Table	**Attribute**	**Type**	**Constraint**
Lineup	id	INT	Primary Key
Lineup	month	INT	NOT NULL
Lineup	date	INT	NOT NULL
Lineup	year	INT	NOT NULL
Lineup	Review flag	BOOLEAN	NOT NULL
Lineup	difficulty	INT	1<=difficulty<=5
Lineup	type	STRING	type = “Postplant”, “Attack”, “Defense”, or “Setup”
Lineup	site	CHAR	site = ‘A’, ‘B’, or ‘C’
Lineup	video	VARCHAR	NOT NULL
Lineup	username	VARCHAR	Foreign Key
Lineup	map_name	VARCHAR	Foreign Key
Player	username	VARCHAR	Primary Key
Player	password	VARCHAR	NOT NULL
Player	Admin	BOOLEAN	NOT NULL
Agent_Tag	lineup_id	VARCHAR	Foreign Key
Agent_Tag	agent_name	VARCHAR	Foreign Key
Ability_Tag	lineup_id	VARCHAR	Foreign Key
Ability_Tag	ability_name	VARCHAR	Foreign Key
Agent	agent_name	VARCHAR	Primary Key
Agent	role	VARCHAR	Role = “Initiator”, “Sentinel”, “Controller”, or

			“Duelist”
Agent	agent_image	BLOB	
Ability	ability_name	STRING	Primary Key
Ability	abiltiy_no	INT	1<=ability_no<=4
Ability	ability_image	BLOB	
Ability	agent_name	STRING	Foreign Key
Map	map_name	STRING	Primary Key
Map	map_image	BLOB	
Rate	lineup_id	INT	Foreign Key
Rate	username	VARCHAR	Foreign Key
Rate	rating	INT	0<=rating<=5

Application Program Design:

```
global name, isAdmin = false

Login()
   name = prompt for username
   password = prompt for password
   if(password == query Player table with name for password attribute)
      loggedIn = true
      isAdmin = query Player table with name for admin attribute
   else
      display error message

LogOut()
   loggedIn = false
   isAdmin = false

SignUp()
   newName = prompt for username
   newPassword = prompt for password
   execute query to add attributes to Player table with newname as username and
newpassword as password (set admin as false)

AddMap()
   if(isAdmin)
      execute query to add attributes to Abiltiy table

AddAgent()
   if(isAdmin)
      execute query to add attributes to Agent table

AddAbility()
   if(isAdmin)
      execute query to add attributes to Ability table

DeletePlayer()
   if(isAdmin)
      specify username
      execute delete query starting at topmost Player table and cascading downwards
```

```
ModifyPlayer()
   if(isAdmin)
      //modify various Player attributes

PostLineup()
   if(loggedIn)
      //prompt for mapName
      //prompt for difficulty
      //prompt for type
      //prompt for site
      //prompt for video file
      //prompt for agent name
      //prompt for ability name
      execute query to add attributes and name to Lineup table
      execute query to add agentName and id attribute to Save table
      execute query to add abilityName and id attribute to Ability table

SaveLineup()
   if(loggedIn)
      //specify lineupID
      execute query to add lineupID and name to Save table

RateLineup()
   if(loggedIn)
      //specify lineupID
      //prompt for rating
      execute query to add lineupID, name, and rating to Rate table
```

Aggregation Functions:

- Get number of posts for a specific user
- Get average rating of a lineup video
- Get lineup video with highest rating

Instructions to setup the database and run the program:

1. Follow instructions here https://www.youtube.com/watch?v=7S_tz1z_5bA&pp=ygUKbXlzcWwgbW9zaA%3D%3D to properly install MySQL and MySQL Workbench and start the server
2. Change your MySQL to access all directories on your pc
3. Copy the ValorantDatabaseAssets folder into the following directory C:\Program Files\MySQL\MySQL Server 8.0
4. In MySQL Workbench, open and run create-valorant-database.sql

5. In a editor, change the host, user, and password to your following details in the database connection part of the code in main.py (labeled with the comment "establish database connection")
6. Run installation.sh If that doesn't work, run installationpip3.sh And if that doesn't work, just run the commands inside the script to install those independencies
7. Run run.sh or main.py using the following command $ python3 main.python3
8. The program should now be running