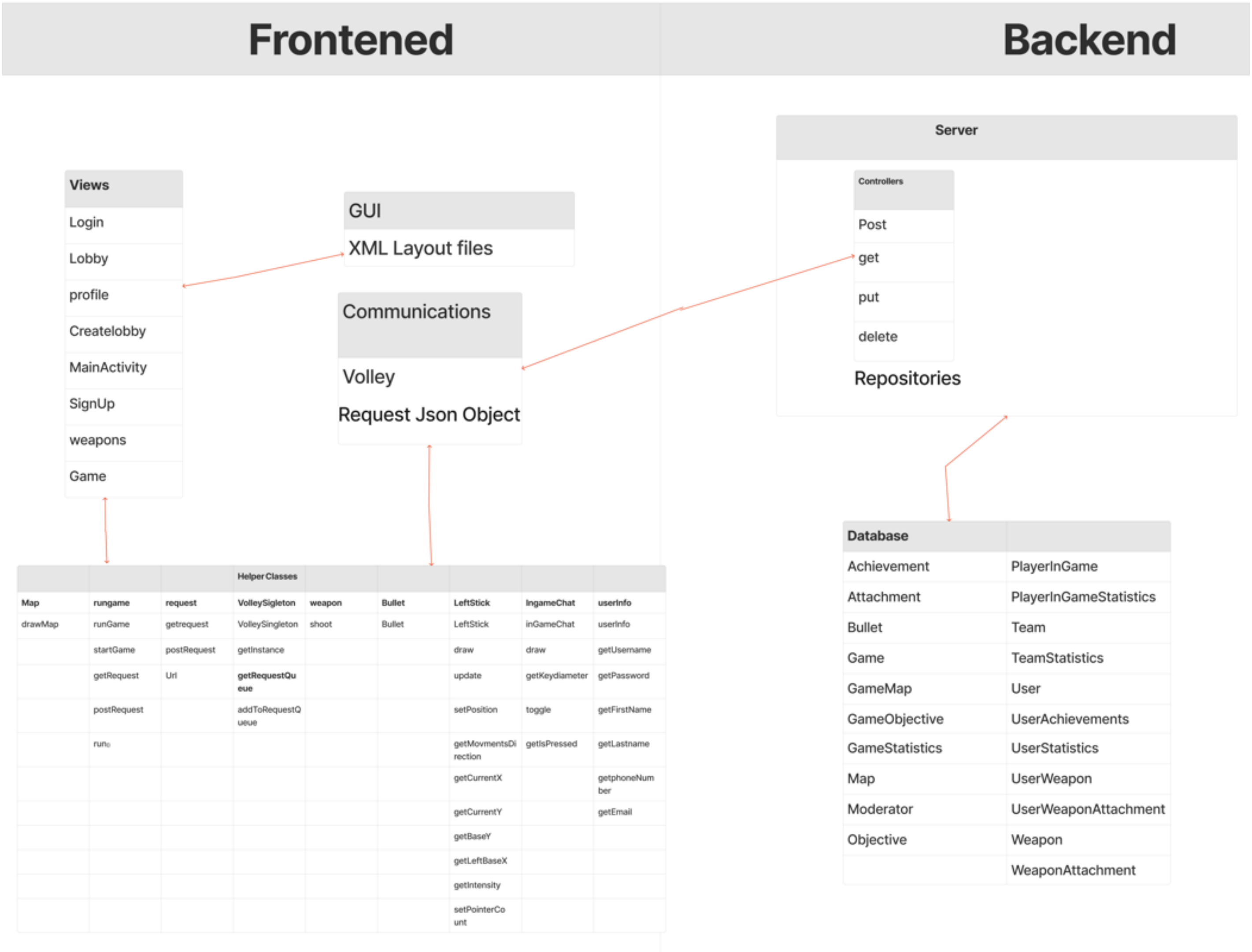
Block Diagram

Team: LM_304
Members:
Breckin Bartels - 25%
Cal Hokanson-Fuchs - 25%
Kat Christofferson - 25%
Varun Jain - 25%

Project Name: CyHawkClash



Frontened

Backend

Views

- Login
- Lobby
- profile
- Createlobby
- MainActivity
- SignUp
- weapons
- Game

GUI

XML Layout files

Communications

Volley

Request Json Object

Server

Controllers

- Post
- get
- put
- delete

Repositories

			Helper Classes					
Map	**rungame**	**request**	**VolleySigleton**	**weapon**	**Bullet**	**LeftStick**	**IngameChat**	**userInfo**
drawMap	runGame	getrequest	VolleySingleton	shoot	Bullet	LeftStick	inGameChat	userInfo
	startGame	postRequest	getInstance			draw	draw	getUsername
	getRequest	Url	**getRequestQueue**			update	getKeydiameter	getPassword
	postRequest		addToRequestQueue			setPosition	toggle	getFirstName
	run()					getMovmentsDirection	getIsPressed	getLastname
						getCurrentX		getphoneNumber
						getCurrentY		getEmail
						getBaseY		
						getLeftBaseX		
						getIntensity		
						setPointerCount		

Database	
Achievement	PlayerInGame
Attachment	PlayerInGameStatistics
Bullet	Team
Game	TeamStatistics
GameMap	User
GameObjective	UserAchievements
GameStatistics	UserStatistics
Map	UserWeapon
Moderator	UserWeaponAttachment
Objective	Weapon
	WeaponAttachment

FRONTEND METHOD DOCUMENTATION

- Bullet: *Create a new bullet object with the following elements:*
    - Angle
    - Current X
    - Current Y
    - Speed
- Game: *Class where all other classes are instantiated and logic/drawing to screen is handled*
    - runGame(game running)
    - LeftStick(left joystick)
    - HealthBar(hp bar)
    - InGameChat(chatbox and logic)
    - Player(player data in game)
    - Map(map to be drawn in game)
- HealthBar: *Class that contains drawing methods for the hp bar*
    - draw() method
    - maxHP
    - currentHP
    - setHP() method
- InGameChat: *Class that contains draw method for chat and logic(not implemented)*
    - *KeyDiameter(diameter for key drawing)*
    - buttonLeft
    - buttonRight
    - buttonTop
    - buttonBottom(variables for drawing toggle button)
    - toggle(boolean for checking if chat should be shown)
    - draw() method
    - toggle() method for toggling chat
- LeftStick: *Stick for handling touch input for movement*
    - leftBaseX, baseY(X/Y location of joystick)
    - CurrentX, CurrentY(X/Y location of touch)
    - intensity(current distance between center of stick and touch(capped at 1)
    - baseRadius(radius of joystick)
    - draw() method
    - update() method(updates angle of stick)
    - setPosition() method(sets current player position)
    - getMovementDirection()(returns angle of stick)
    - getCurrentX(),getCurrentY(),getLeftBaseX(),getBaseY(),getIntensity()(self-explanatory)

- MainActivity: *Handles creation of Game class and setting screen to it.*
    - onCreate() does above^
- Map: *Draws map*
    - drawMap()(draws map)
- PlayerInGame: *Stores data of player while in game*
    - drawPlayer()(draws player)
    - takeDamage()(damages player)
    - respawn()(respawns player)
    - update(updates location)
    - getLocation(),getHealth(),setPlayerTeam(),setPlayerID(),getPlayerID(),getPlayerX(),getPlayerY()(Does as they say)
- runGame: *Runs the game and updates the game multiple times per second*
    - run() does above^
    - putRequest()(updates json files)
    - getRequest()(receives json files)
    - startGame()(starts game)
- Weapon: *Shoots gun/creates bullets, not implemented yet*
    - magazine
    - currentBullets
    - shoot()(not implemented)

- Lobby: *Where the player can join other people’s lobbies, and Go to create their own lobbies.*
  - Button CreatLobbyB
  - SetOnClickListener(new View onclickLitsener): when create Lobby is clicked
  - onclick(): Opens a new activity to Create lobby
  - Spinner Dropdown
  - String [] Items =setting, Profile, weapons, Logout: each of the items in the drop down. When chosen it takes the plates to the activity.
  - ArrayAdapter adapter: Create the function of a drop down spinner
  - setItemSelected(AdapterView<?> parent, View view, int position, long id) litsents for when an item is clicked.
  - onNothingSelected(AdapterView<?> parent) does nothing when no item is selected.
- Profile: *Where the user can see their information, and make changes to their username, and password.*
  - Spinner dropdown2:
  - String [] Items =Back, Lobby Logout: each of the items in the drop down. When chosen it takes the plates to the activity.
  - ArrayAdapter adapter: Create the function of a drop down spinner
  - setItemSelected(AdapterView<?> parent, View view, int position, long id) litsents for when an item is clicked.
  - onNothingSelected(AdapterView<?> parent) does nothing when no item is selected.
- SignUp: *Where the player sign up for an account for the game*
  - ArrayList<userInfo> userInfos
  - String username,
  - string password,
  - String thefirst,
  - Sting theLast,
  - String theemail.
  - String thephone
  - Button SignupCreate
  - EditText newusersanem
  - EditText Newpassword
  - EditText confirmpassword
  - EditTes firN
  - EditText lastN
  - EditText email
  - EditText phoneN

  - String id
  - SetOnCickListener(new Viwe onClickLitsener):; when create is clicked it creates a new user and sends it to the backend.
  - onclik (View view)
  - Button backtoLogin
  - SetOnCickListener(new Viwe onClickLitsener): when back is clicked take you back to the username and password activity
  - Spinner dropdown5
  - String [] Items =team, cyclone, Hawkes: when players choose a team they get a default profile picture according to the team.
  - ArrayAdapter adapter: Create the function of a drop down spinner
  - setItemSelected(AdapterView<?> parent, View view, int position, long id) litsents for when an item is clicked.
  - onNothingSelected(AdapterView<?> parent) does nothing when no item is selected.
  - postRequest(String path)
- Weapon_viiew: Beagle to view all the weapons that the player own, and the weapons attachments.
  - JsaonArrayRequest weapoes; the weapon [ost request
  - JsonArrayRequest attach: the attachment post request
  - Text view wep ; the view for list of weapons
  - String Wep_id; the weapon id
  - TextView addAtts: the view for the attachments.
  - postRequest(string path): post request for weapons
  - getAttachmentAray(String path)get request for list of attachments
  - PosrequaetAtt(String path): post request for list of attachments;

- userInfo: *This is the class where the user is created*
    - String userName
    - String passWord
    - String firstName
    - String lastName
    - String PhoneNumber
    - String email
    - userInfo(String userName, string passWord, String the FirstName, Sting LastNem, String Email. String Phonenumber)
    - getusername()
    - getPasswrod()
    - getFirstName()
    - getLastName()
    - getEamil()
    - getPhoneNumber()
- VolleySingleton: *Where the communication happens the get and post request get put in a queue.*
    - VolleySingleton instance
    - RequestQueue requestQueue
    - ImageLoader imageLoader
    - VolleySingleton (context context)
    - getInstance(Context Context)
    - getRequestQueue()
    - addToRequestQueue(Request<T> req)
    - getInamgeLoader()
- Weapon: *Where the player can see what weapons they own, and be able to upgrade them*
    - Spinner dropdown3:
    - String [] Items =Back, Lobby Logout: each of the items in the drop down. When chosen it takes the plates to the activity.
    - ArrayAdapter adapter: Create the function of a drop down spinner
    - setItemSelected(AdapterView<?> parent, View view, int position, long id) litsents for when an item is clicked.
    - onNothingSelected(AdapterView<?> parent) does nothing when no item is selected.

- CreateLobby: *Where the players can create their own lobbies*
    - Spinner dropdown4:
    - String [] Items =Back, Lobby Logout: each of the items in the drop down. When chosen it takes the plates to the activity.
    - ArrayAdapter adapter: Create the function of a drop down spinner
    - setItemSelected(AdapterView<?> parent, View view, int position, long id) litsents for when an item is clicked.
    - onNothingSelected(AdapterView<?> parent) does nothing when no item is selected.

BACKEND

Controllers

Our controllers send, update, get, and delete information from the database based on information that has been given.

- **POST:** Transmit information pertaining to an item that should be appended to the database.
- **GET:** Request information, often with an accompanying identifier to locate the specific item of interest within the database.
- **PUT:** Send data with the intention of altering a specific item within the database.
- **DELETE:** Send an identifier to softly delete a specific item from the database.

Repositories

The repositories for each table in the database contains findBy methods to retrieve specific instances from the database.

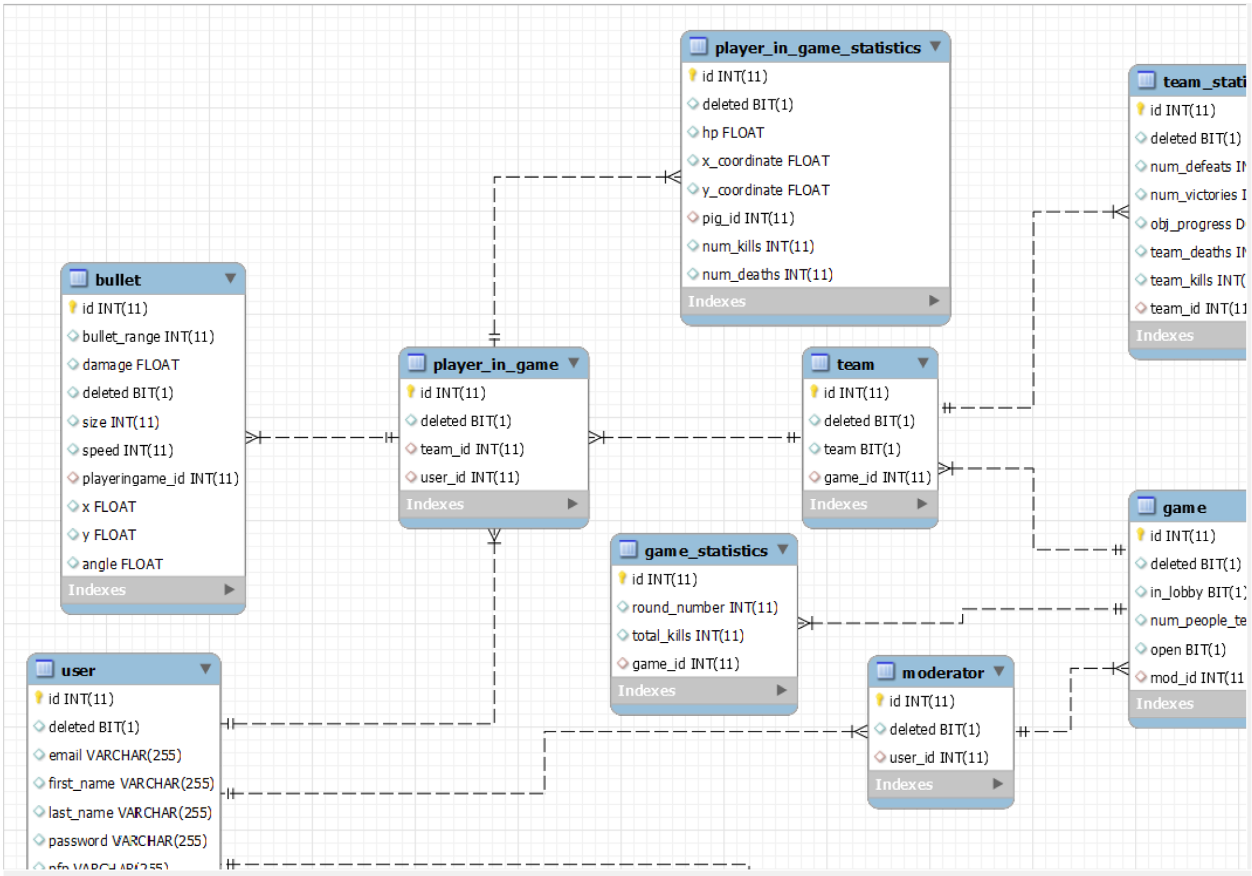
DATABASE:

- *Achievement*:
  - This table will contain Achievements that the user can complete in order to get rewards.
- *Attachment:*
  - These are attachments with stats that will be available for a weapon.
- *Bullet:*
  - This table tracks all bullets that have been fired in all games.
- *Game:*
  - This table contains information about all games that have or are being played.
- *GameMap:*
  - This table keeps track of the Map status for each game.
- *GameObjective:*
  - This table stores information as to which objective is being used for a game.
- *GameStatistics:*
  - This table stores all statistical information regarding each game.
- *Map:*
  - This table will contain static maps, this will contain an objective type, and a string that is the map.
- *Moderator:*
  - This will store moderators of lobbies.
- *Objective:*
  - This will store all possible objectives a game can choose from.

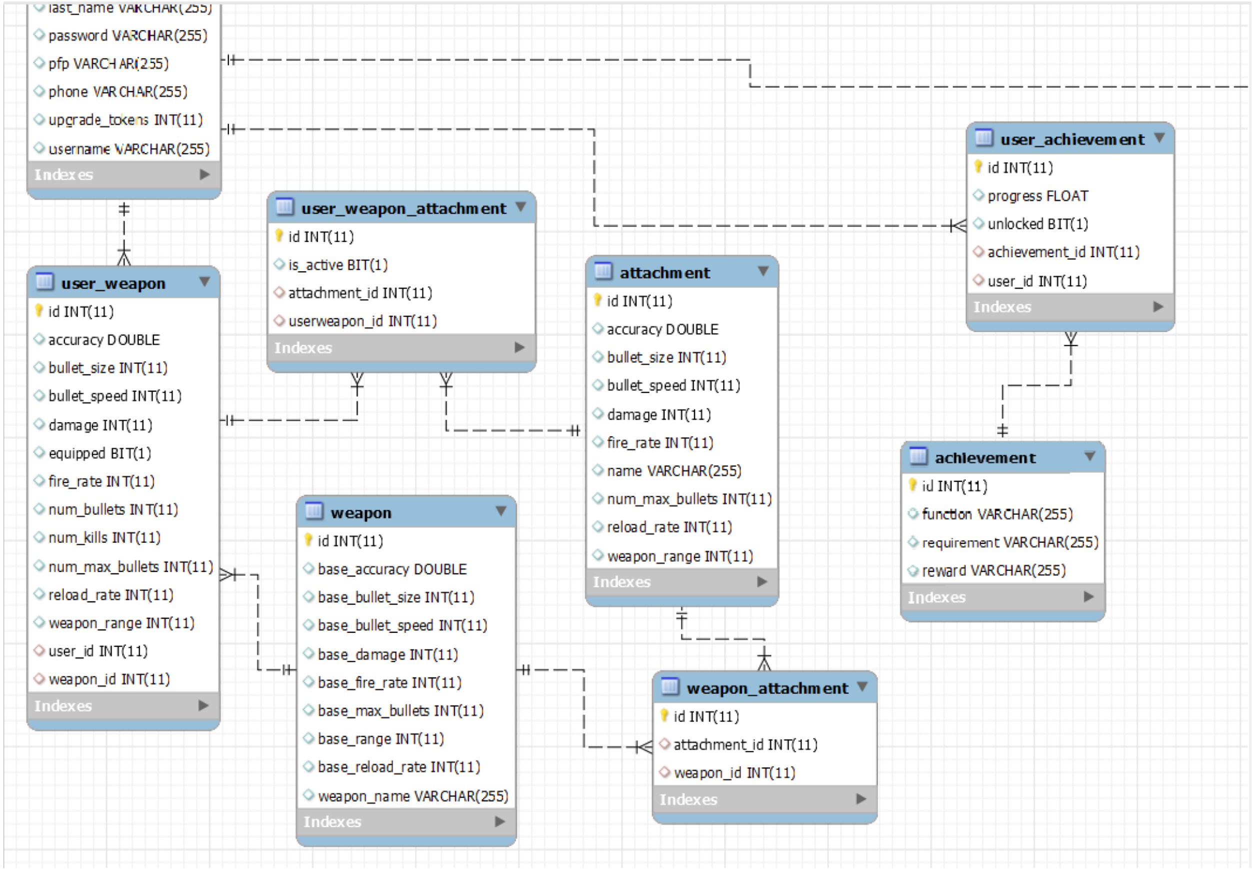
- *PlayerInGame:*
    - A playerInGame will be made when a game is started. This will contain the team they are on and their user.
- *PlayerInGameStatistics:*
    - This table will contain x, y coordinates, health, and number of kills for the playerInGame
- *Team:*
    - This table contains game information that is specific to the team.
- *TeamStatistics:*
    - This table contains team statistics such as number of kills or number of deaths.
- *User:*
    - This contains all user informations
- *UserAchievements:*
    - This table contains achievements that are available to the user
- *UserStatistics:*
    - This table contains user stats from all of their games.
- *UserWeapon:*
    - This table contains Weapons that are available to the user, and whether or not that weapon is being used.
- *UserWeaponAttachment:*
    - This table contains a list of attachments available for each UserWeapon, whether or not it is being used.
- *Weapon:*
    - Static list of Weapons available in the game.
- *WeaponAttachment:*
    - Static list of Weapon Attachments available in the game.

Database Diagram

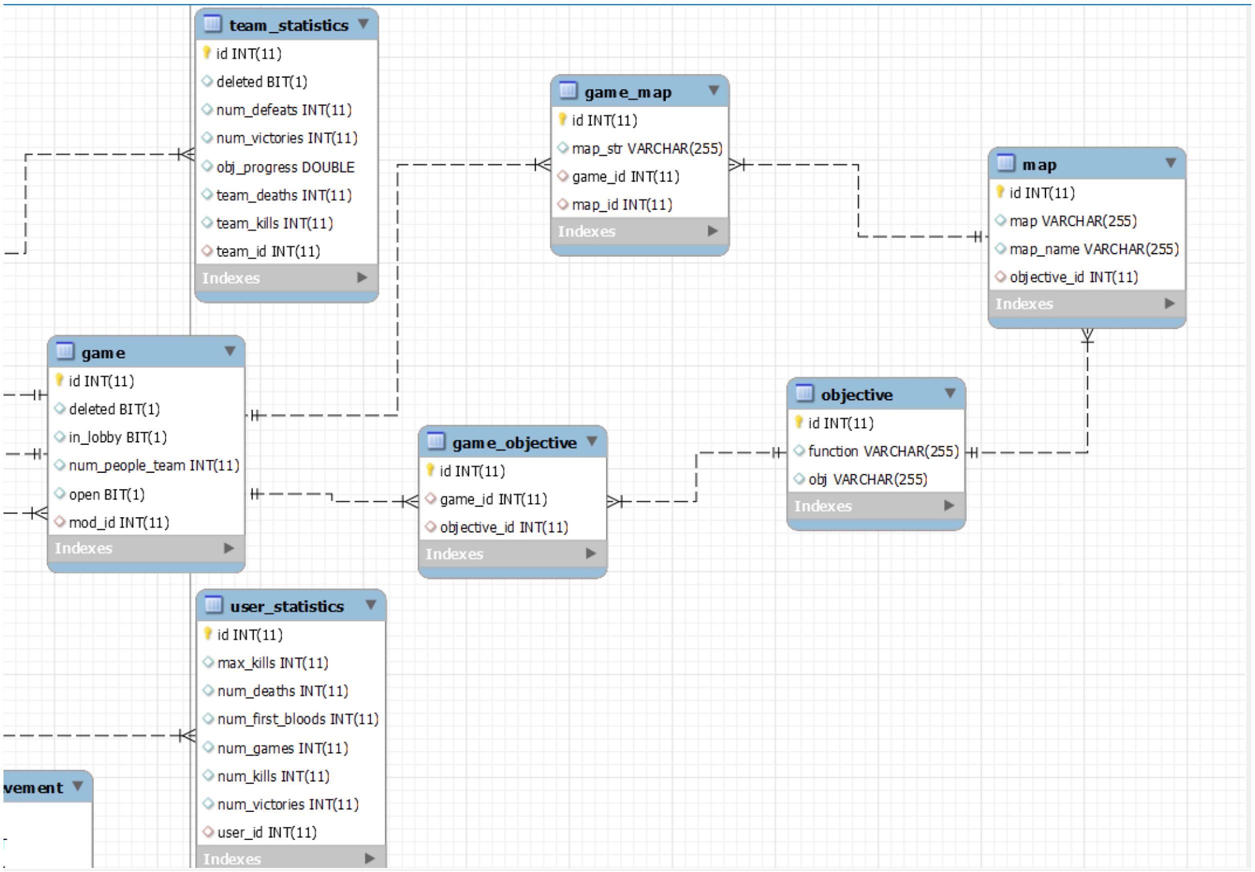
Top-left



Bottom-left


last_name VARCHAR(255)
password VARCHAR(255)
pfp VARCHAR(255)
phone VARCHAR(255)
upgrade_tokens INT(11)
username VARCHAR(255)
Indexes
user_weapon_attachment
id INT(11)
is_active BIT(1)
attachment_id INT(11)
userweapon_id INT(11)
Indexes
user_achievement
id INT(11)
progress FLOAT
unlocked BIT(1)
achievement_id INT(11)
user_id INT(11)
Indexes
user_weapon
id INT(11)
accuracy DOUBLE
bullet_size INT(11)
bullet_speed INT(11)
damage INT(11)
equipped BIT(1)
fire_rate INT(11)
num_bullets INT(11)
num_kills INT(11)
num_max_bullets INT(11)
reload_rate INT(11)
weapon_range INT(11)
user_id INT(11)
weapon_id INT(11)
Indexes
attachment
id INT(11)
accuracy DOUBLE
bullet_size INT(11)
bullet_speed INT(11)
damage INT(11)
fire_rate INT(11)
name VARCHAR(255)
num_max_bullets INT(11)
reload_rate INT(11)
weapon_range INT(11)
Indexes
achievement
id INT(11)
function VARCHAR(255)
requirement VARCHAR(255)
reward VARCHAR(255)
Indexes
weapon
id INT(11)
base_accuracy DOUBLE
base_bullet_size INT(11)
base_bullet_speed INT(11)
base_damage INT(11)
base_fire_rate INT(11)
base_max_bullets INT(11)
base_range INT(11)
base_reload_rate INT(11)
weapon_name VARCHAR(255)
Indexes
weapon_attachment
id INT(11)
attachment_id INT(11)
weapon_id INT(11)
Indexes

Top-right



team_statistics
- id INT(11)
- deleted BIT(1)
- num_defeats INT(11)
- num_victories INT(11)
- obj_progress DOUBLE
- team_deaths INT(11)
- team_kills INT(11)
- team_id INT(11)
- Indexes

game_map
- id INT(11)
- map_str VARCHAR(255)
- game_id INT(11)
- map_id INT(11)
- Indexes

map
- id INT(11)
- map VARCHAR(255)
- map_name VARCHAR(255)
- objective_id INT(11)
- Indexes

game
- id INT(11)
- deleted BIT(1)
- in_lobby BIT(1)
- num_people_team INT(11)
- open BIT(1)
- mod_id INT(11)
- Indexes

objective
- id INT(11)
- function VARCHAR(255)
- obj VARCHAR(255)
- Indexes

game_objective
- id INT(11)
- game_id INT(11)
- objective_id INT(11)
- Indexes

user_statistics
- id INT(11)
- max_kills INT(11)
- num_deaths INT(11)
- num_first_bloods INT(11)
- num_games INT(11)
- num_kills INT(11)
- num_victories INT(11)
- user_id INT(11)
- Indexes

Link to the MySQL workbench diagram version: project.mwb

It will be easier to view and understand the connections between tables.