CPSC 304 Project Cover Page

Milestone #: ___2___

Date: ___Oct.20, 2023__

Group Number: __46______

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Vito Lin	76174770	x0n3w	vitolin2003@gmail.com
Yuang Chen	84122456	m9s0n	bryan1mageacal@gmail.com
Corina Dong	18129544	j5p6x	corina.dong@gmail.com

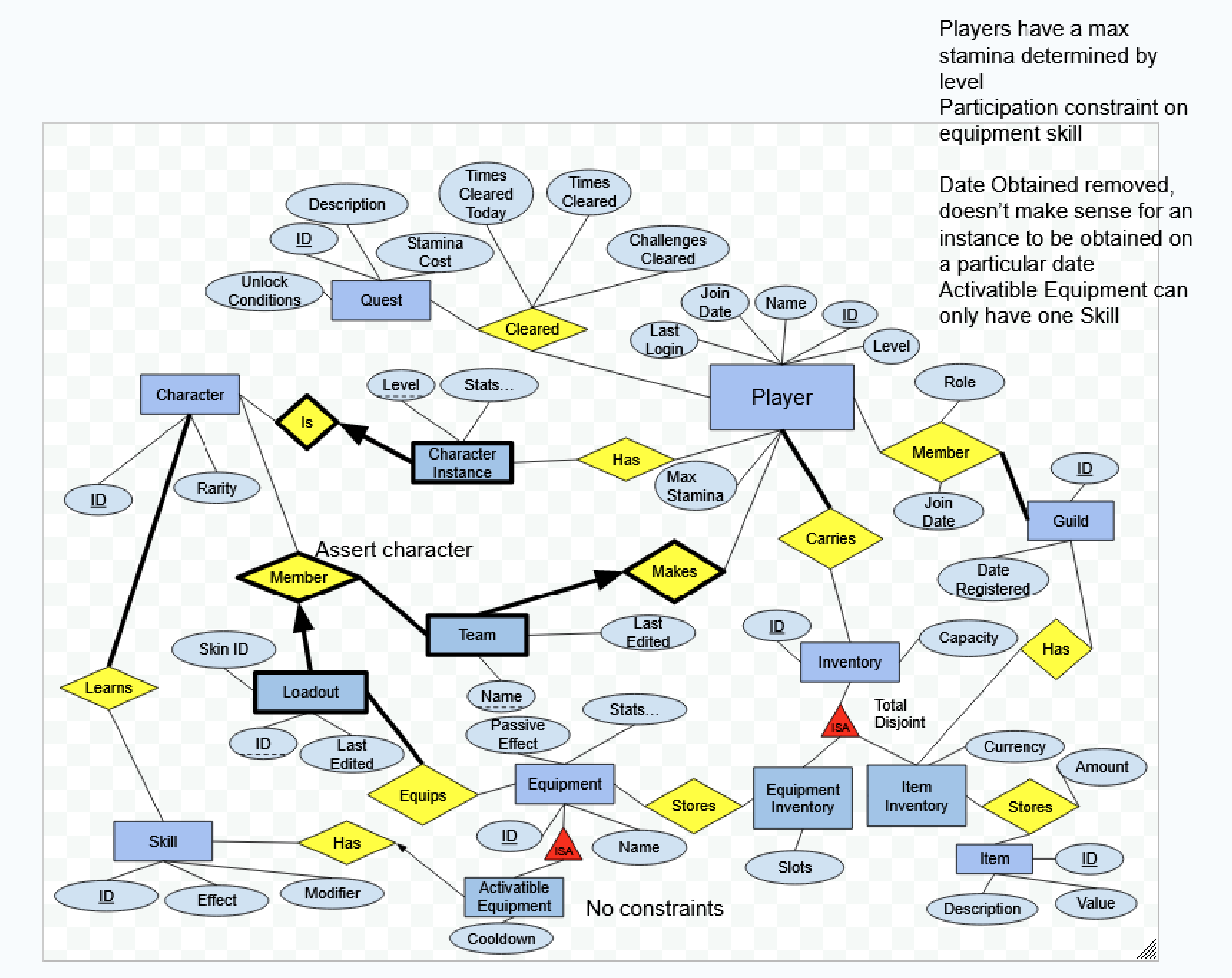
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

Summary:

The aim of this application is to track player progress for a generic gacha game. This project will support future gacha game creation by providing a complete framework for any gacha game project. As well, it should also provide the ability for an administrator to edit player data manually, as well as store object information, such as items and characters so that the game can retrieve the data and display the information.

ER Diagram:



Players have a max stamina determined by level
Participation constraint on equipment skill

Date Obtained removed, doesn't make sense for an instance to be obtained on a particular date
Activatible Equipment can only have one Skill

Assert character

Total Disjoint

No constraints

Schema:

```
Player(PRIMARY KEY ID: int, Name:varchar, MaxStamina: int, Level:int, JoinDate: date, LastLogin: date, FK Guild: guild.id, FK Inventory1: inventory.id NOT NULL);

Quest(PRIMARY KEY ID: int, StaminaCost: int, UnlockConditions:int, Description: varchar)
ClearedQuests(PRIMARY KEY (FK QuestId: quest.id, FK PlayerID: player.id))


Inventory(PK ID: int, Capacity: int)
PlayerInventories(PK(FK Player.ID, FK Inventory.id))

Skill(PRIMARY KEY ID, effect: VARCHAR, modifier: INT)
Character(PRIMARY KEY ID: int, CK Name: varchar, Rarity: int, FOREIGN KEY skill1: skill.ID NOT NULL)
CharacterSkills(PRIMARY KEY(FK character.id, FK skill.id))

Equipment(PRIMARY KEY ID: int, CK name(varchar), passive effect (varchar), hp: int, str: int, def: int)
ActivatibleEquipment(PK FK Equipment.ID, FK Skill.ID, Cooldown: int)

EquipmentInventory(PK FK Inventory.id, slots: int)
StoredEquipment(PK(FK EquipmentInventory.id, FK Equipment.id))

Item(PK ID: int, description: varchar, value: int)
ItemInventory(PK FK Inventory.id, currency: int)
StoredItems(PK(FK Inventory.id, FK Item.ID), amount: int)

CharacterInstance(PK(FK Character.ID, Level: int), HP: int, Str: int, def: int)
Team(PK(FK Player.id, Name: varchar), LastEdited: date)
Loadout(PK(ID, FK Character.ID, FK PlayerID, FK Team.Name), SkinID: int, LastEdited: date)
>>>>LoadoutEquipment(PK(Loadout.ID, FK Character.ID, FK PlayerID, FK Team.Name, Equipment.id))
TeamMembers(PK(FK PlayerID, FK Team.Name, FK Character.ID, FK Loadout.id))

>>OwnedCharacters(PK(Player.ID, Character.ID, CharacterInstance.Level))

Guild(PRIMARY KEY ID: int, DateRegistered: date), FOREIGN KEY leader:player ID))
GuildMembers(PK(FK Guild.ID, FK Player.id), role: int, join date: date)
GuildInventories(PK(FK Guild.ID, FK Inventory.id))
```

FDS and Normalization
Player.ID->Name, Join Date, Last Login, Level, Max Stamina, Guild
Level -> Max Stamina
Team.Name, Player.ID -> Last Edited
Character.Id -> Rarity
Character.ID, Character.Level -> CharacterInstance.Stats
Quest.ID -> Description, Unlock Conditions, Stamina Cost
Player.ID, Quest.ID -> Description, Unlock Conditions, Stamina Cost
Guild.ID -> Date Registered
Item.ID -> Value, Description
ItemInventory.ID, Item.ID -> Amount
ItemInventory.ID -> Currency, Capacity
EquipmentInventory.ID -> Slots, Capacity
Equipment.ID -> Passive Effect, Stats…, Name
ActivatibleEquipment.ID -> Passive Effect, Stats…, Name, cooldown, Skill
Loadout.ID, Character.id -> Skin ID, Loadout.Last Edited
Skill.ID -> Effect, Modifier
Team.Name, Player.Id -> Team.Last Edited

Normalizing Player(PRIMARY KEY ID, Name:varchar, Level:int, MaxStamina: int, JoinDate: date, LastLogin: date, FK Guild: guild.id, FK Inventory1: inventory.id NOT NULL) where Level -> MaxStamina into 3NF/BCNF:
Level is not a key, so Level->MaxStamina violates BCNF. Splitting into two tables:
Player(PRIMARY KEY ID, Name:varchar, FK Level:int, JoinDate: date, LastLogin: date, FK Guild: guild.id, FK Inventory1: inventory.id NOT NULL)
PlayerStaminas(PRIMARY KEY Level:int, MaxStamina: int)

Final Schema:
Player(PRIMARY KEY ID, Name:varchar, FK Level:int, JoinDate: date, LastLogin: date, FK Guild: guild.id, FK Inventory1: inventory.id NOT NULL)
PlayerStaminas(PRIMARY KEY Level:int, MaxStamina: int)
LastLogin: date, FK Guild: guild.id, FK Inventory1: inventory.id NOT NULL);

Quest(PRIMARY KEY ID: int, StaminaCost: int, UnlockConditions:int, Description: varchar)
ClearedQuests(PRIMARY KEY (FK QuestId: quest.id, FK PlayerID: player.id))

Inventory(PK ID: int, Capacity: int)
PlayerInventories(PK(FK Player.ID, FK Inventory.id))

Skill(PRIMARY KEY ID, effect: VARCHAR, modifier: INT)
Character(PRIMARY KEY ID: int, CK Name: varchar, Rarity: int, FOREIGN KEY skill1: skill.ID NOT NULL)
CharacterSkills(PRIMARY KEY(FK character.id, FK skill.id))

Equipment(PRIMARY KEY ID: int, CK name(varchar), passive effect (varchar), hp: int, str: int, def: int)
ActivatibleEquipment(PK FK Equipment.ID, FK Skill.ID, Cooldown: int)

EquipmentInventory(PK FK Inventory.id, slots: int)
StoredEquipment(PK(FK EquipmentInventory.id, FK Equipment.id))

Item(PK ID: int, description: varchar, value: int)
ItemInventory(PK FK Inventory.id, currency: int)
StoredItems(PK(FK Inventory.id, FK Item.ID), amount: int)

CharacterInstance(PK(FK Character.ID, Level: int), HP: int, Str: int, def: int)
Team(PK(FK Player.id, Name: varchar), LastEdited: date)
Loadout(PK(ID, FK Character.ID, FK PlayerID, FK Team.Name), SkinID: int, LastEdited: date)
>>>>LoadoutEquipment(PK(Loadout.ID, FK Character.ID, FK PlayerID, FK Team.Name, Equipment.id))
TeamMembers(PK(FK PlayerID, FK Team.Name, FK Character.ID, FK Loadout.id))

>>OwnedCharacters(PK(Player.ID, Character.ID, CharacterInstance.Level))

Guild(PRIMARY KEY ID: int, DateRegistered: date), FOREIGN KEY leader:player ID))
GuildMembers(PK(FK Guild.ID, FK Player.id), role: int, join date: date)
GuildInventories(PK(FK Guild.ID, FK Inventory.id))

```
CREATE TABLE Player (
        ID INT PRIMARY KEY,
        Name VARCHAR,
        Level INT,
        JoinDate DATE,
        LastLogin DATE,
        Inventory1 INT,
        FOREIGN KEY (Level ) REFERENCES PlayerStaminas (Level),
        FOREIGN KEY (Inventory1) REFERENCES Inventory(ID)
);

CREATE TABLE PlayerStaminas (
        Level INT PRIMARY KEY,
        MaxStamina INT,
);


CREATE TABLE Quest (
        ID INT PRIMARY KEY,
        StaminaCost INT,
        UnlockConditions INT,
        Description VARCHAR
);
CREATE TABLE ClearedQuests (
        QuestId INT,
        PlayerID INT,
        PRIMARY KEY (QuestId, PlayerID),
        FOREIGN KEY (QuestId) REFERENCES Quest(ID),
        FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);

CREATE TABLE Inventory (
        ID INT PRIMARY KEY,
        Capacity INT
);
CREATE TABLE PlayerInventories (
        PlayerID INT,
        InventoryID INT,
        PRIMARY KEY (PlayerID, InventoryID),
        FOREIGN KEY (PlayerID) REFERENCES Player(ID),
        FOREIGN KEY (InventoryID) REFERENCES Inventory(ID)
);
```

```
CREATE TABLE Skill (
	ID INT PRIMARY KEY,
	Effect VARCHAR,
	Modifier INT
);
CREATE TABLE Character (
	ID INT PRIMARY KEY,
	Name VARCHAR NOT NULL UNIQUE,
	Rarity INT,
	Skill1 INT NOT NULL,
	FOREIGN KEY (Skill1) REFERENCES Skill(ID)
);
CREATE TABLE CharacterSkills (
	CharacterID INT,
	SkillID INT,
	PRIMARY KEY (CharacterID, SkillID),
	FOREIGN KEY (CharacterID) REFERENCES Character(ID),
	FOREIGN KEY (SkillID) REFERENCES Skill(ID)
);

CREATE TABLE Equipment (
	ID INT PRIMARY KEY,
	Name VARCHAR,
	PassiveEffect VARCHAR,
	Stats VARCHAR
);
CREATE TABLE ActivatibleEquipment (
	EquipmentID INT,
	SkillID INT,
	Cooldown INT,
	PRIMARY KEY (EquipmentID, SkillID),
	FOREIGN KEY (EquipmentID) REFERENCES Equipment(ID),
	FOREIGN KEY (SkillID) REFERENCES Skill(ID)
);
CREATE TABLE EquipmentInventory (
	InventoryID INT,
	Slots INT,
	PRIMARY KEY (InventoryID),
	FOREIGN KEY (InventoryID) REFERENCES Inventory(ID)
);
CREATE TABLE StoredEquipment (
	EquipmentID INT,
	InventoryID INT,
	PRIMARY KEY (EquipmentID, InventoryID),
	FOREIGN KEY (EquipmentID) REFERENCES Equipment(ID),
	FOREIGN KEY (InventoryID) REFERENCES Inventory(ID)
);

CREATE TABLE Item (
	ID INT PRIMARY KEY,
	Description VARCHAR,
	Value INT
);
CREATE TABLE ItemInventory (
	InventoryID INT,
	Currency INT,
	PRIMARY KEY (InventoryID),
	FOREIGN KEY (InventoryID) REFERENCES Inventory(ID)
);
CREATE TABLE StoredItems (
	InventoryID INT,
	ItemID INT,
	Amount INT,
	PRIMARY KEY (InventoryID, ItemID),
	FOREIGN KEY (InventoryID) REFERENCES Inventory(ID),
	FOREIGN KEY (ItemID) REFERENCES Item(ID)
);

CREATE TABLE CharacterInstance (
	CharacterID INT,
	Level INT,
	HP INT,
	Str INT,
	Def INT,
	PRIMARY KEY (CharacterID, Level),
	FOREIGN KEY (CharacterID) REFERENCES Character(ID)
);
CREATE TABLE Team (
	PlayerID INT,
	Name VARCHAR,
	LastEdited DATE,
	PRIMARY KEY (PlayerID, Name),
	FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);
CREATE TABLE Loadout (
	ID INT,
	PlayerID INT,
	CharacterID INT,
        TeamName VARCHAR,
        SkinID INT,
        LastEdited DATE,
        Equipped INT,
        PRIMARY KEY (ID, CharacterID, TeamName),
        FOREIGN KEY (CharacterID) REFERENCES Character(ID),
        FOREIGN KEY (Equipped) REFERENCES Equipment(ID),

        FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);
CREATE TABLE TeamMembers (
        PlayerID INT,
        TeamName VARCHAR,
        CharacterID INT,
        LoadoutID INT,
        PRIMARY KEY (PlayerID, TeamName, CharacterID, LoadoutID),
        FOREIGN KEY (CharacterID) REFERENCES Character(ID)
        FOREIGN KEY (LoadoutID) REFERENCES Loadout(ID)
        FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);
CREATE TABLE LoadoutEquipment (
        LoadoutID INT,
        CharacterID INT,
        PlayerID INT,
        TeamName VARCHAR,
        EquipmentID INT,
        PRIMARY KEY (PlayerID, LoadoutID INT, CharacterID, TeamName),
        FOREIGN KEY (LoadoutID) REFERENCES Loadout(ID)
        FOREIGN KEY (CharacterID) REFERENCES Character(ID)
        FOREIGN KEY (PlayerID) REFERENCES Player(ID)
)

CREATE TABLE Guild (
        ID INT PRIMARY KEY,
        DateRegistered DATE,
        Leader INT,
        FOREIGN KEY (Leader) REFERENCES Player(ID)
);
CREATE TABLE GuildMembers (
        GuildID INT,
        PlayerID INT,
        Role INT,
        JoinDate DATE,
        PRIMARY KEY (GuildID, PlayerID),
	FOREIGN KEY (GuildID) REFERENCES Guild(ID),
	FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);
CREATE TABLE GuildInventories (
	GuildID INT,
	InventoryID INT,
	PRIMARY KEY (GuildID, InventoryID),
	FOREIGN KEY (GuildID) REFERENCES Guild(ID),
	FOREIGN KEY (InventoryID) REFERENCES Inventory(ID)
);
```

```
INSERT INTO Player (ID, Name, Level, JoinDate, LastLogin, Guild, Inventory1)
VALUES (1001, 'Player1', 10, '2023-01-01', '2023-01-02', 1, 10),
          (1002, 'Player2', 0, '2023-01-02', '2023-01-03', 1, 20),
          (1003, 'Player3', 1, '2023-01-03', '2023-01-04', 2, 30),
          (1004, 'Player4', 2, '2023-01-04', '2023-01-05', NULL, 40),
          (1005, 'Player5', 3, '2023-01-05', '2023-01-06', 3, 50);

INSERT INTO PlayerStaminas(Level,

INSERT INTO Quest (ID, StaminaCost, UnlockConditions, Description)
VALUES (1, 0, NULL, “Tutorial”),
          (2, 0, 2, “Prologue”),
          (3, 5, 3, “Forest Entrance”),
          (4, 8, 4, “Forest Clearing'),
          (5, 10, 5, “Deep Forest);
INSERT INTO ClearedQuests (QuestId, PlayerID)
VALUES (1, 1001), (1, 1002), (1, 1003), (1, 1004), (1, 1005);

INSERT INTO Inventory (ID, Capacity)
VALUES (10, 100), (20, 50), (30, 50), (40, 50), (50, 50), (11, 100), (21, 50), (31, 50), (41, 50),
(51, 50),
INSERT INTO PlayerInventories (PlayerID, InventoryID)
VALUES (1001, 10), (1002, 20), (1003, 30), (1004, 40), (1005, 50), (1001, 11), (1002, 21),
(1003, 31), (1004, 41), (1005, 51);

INSERT INTO Skill (ID, Effect, Modifier) VALUES
(1, “Basic Attack”, 100),
(2, “Skill”, 300),
(3, “Ultimate”, 400),
(4, “Water Stream”, 200),
(5, “Lapidus Domini”, 600);

INSERT INTO Character (ID, Name, Rarity, Skill1) VALUES
(1, “Protagonist”, 5, 1),
(2, “Zhong”, 3, 5),
(3, “Luka”, 4, 2),
(4, “Mercury”, 5, 4),
(5, “Arc, 5, 3);
INSERT INTO CharacterSkills (CharacterID, SkillID) VALUES
(1, 1),
(2, 5),
(3, 2),
(4, 4),
(5, 3);
```

```
INSERT INTO Equipment (ID, Name, Str, PassiveEffect)
VALUES (1, “Aquila Favonia”, 1000, “Damage Reflect”), (2, “Dull Blade”, 0, “Fragile”), (3,
“Hanatsuki Paddle”, 100, “Shield”), (4, “Rusty Sickle”, 10, NULL), (5, “Jotunheim”, 5000,
“Endless Torrent”), (6, “Jar”, 0, “Fragile”), (7, “Stick”, “Fragile”);

INSERT INTO ActivatibleEquipment(ID, Cooldown)
VALUES (1, 30), (3, 30), (5, 15), (6, 0), (7, 0);
INSERT INTO StoredEquipment (EquipmentID, InventoryID) VALUES
(11, 1),
(11, 2),
(11, 3);
(21, 1);
(31, 1);



INSERT INTO Item(ID, Description, Value)
VALUES (1, “Attorney’s Badge”, 0), (2, “Apple”, 1), (3, “Sus Steak”, 0), (4, “Emergency Rations”,
100), (5, “Small Gemstone”, 5000)
INSERT INTO ItemInventory(ID, Currency)
VALUES (10, 1000), (20, 1), (30, 1), (40, 1), (50, 1);
INSERT INTO EquipmentInventory(ID, Slots)
VALUES (11, 1), (21, 1), (31, 1), (41, 1), (51, 1);



INSERT INTO StoredItems (InventoryID, ItemID, Amount)
VALUES
(10, 1, 1),
(10, 2, 10),
(10, 3, 1),
(20, 2, 20),
(30, 4, 1);
```

```
INSERT INTO CharacterInstance (CharacterID, Level, HP, Str, Def) VALUES
(1, 10, 1000, 100, 100),
(2, 20, 2000, 200, 200),
(3, 30, 3000, 300, 300);
```