

## 4402 Database Design Report

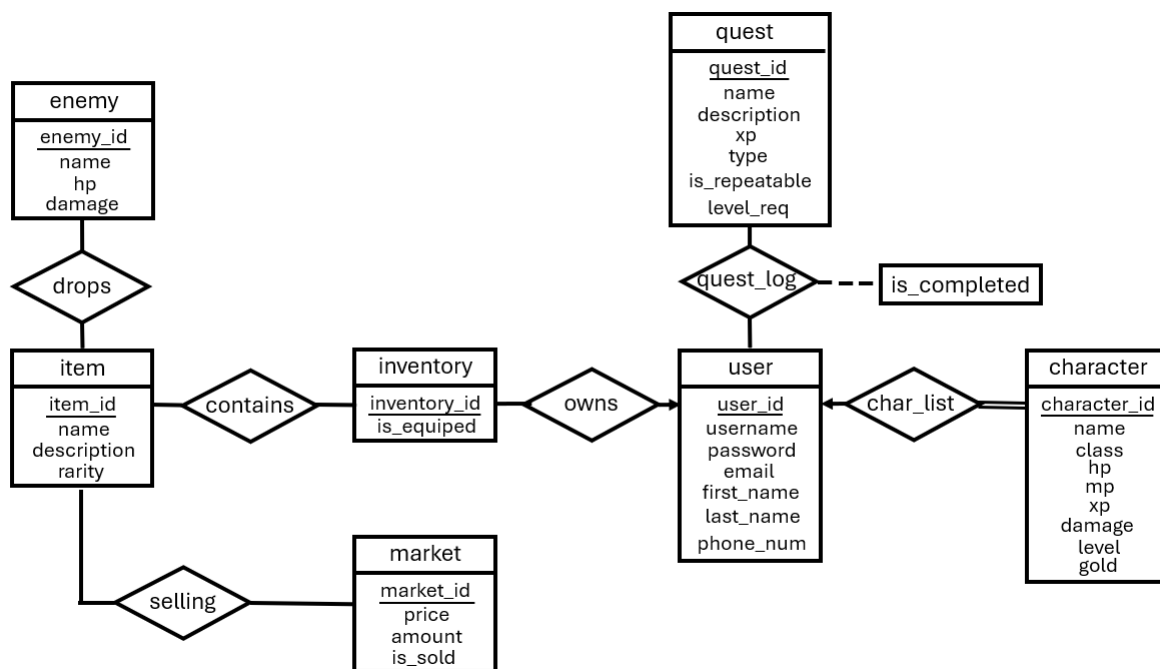
### Participation:

This was a solo project from Jacob Kinchen

### Enterprise Description:

The enterprise chosen for this project was an online role-playing game (rpg) titled "Quest Unlimited". In Quest Unlimited, players can register accounts, manage characters and items, shop at a marketplace, fight enemies and complete quests for rewards. A database is needed for this enterprise because user data, especially tied to something like a video game that people could put a lot of time into, is important to manage properly not only for the performance of your game but also to avoid losing data for players that have put a lot of time into the game. Especially when your game handles thousands of users. The data encapsulated by the database includes user information, character management, quest tracking, item and inventory management, combat, and market operations.

### ER Diagram:



### Relationship Schema:

- Primary Keys: ■
- **owns**(user\_id, inventory\_id)
  - Relationship between the player and their inventory. Allows tracking which inventory belongs to which user.
- **selling**(item\_id, market\_id)

- Shows what the items are being sold at the market.
- **drops**(item\_id, enemy\_id, amount, xp, drop\_chance)
  - When an enemy is defeated, it has an associated item that it will drop at a certain percentage as well as an amount of xp.
- **quest\_log**(user\_id, quest\_id, is\_completed)
  - Allows the user to view a log of all of their quest and if they've completed them or not.
- **char\_list**(user\_id, character\_id)
  - A list of all of the characters a user has.
- **contains**(inventory\_id, item\_id, amount)
  - Link for items in inventories and allows for the declaration of how many occurrences of that item in the inventory.

### User Interface:

A simple command line interface (CLI) was chosen for this project because building a whole GUI seemed out of scope for this project. There are a few different points of interaction between the user and the application:

- **Welcome:**

The user is welcomed and asked whether they want to **Log In** or **Create An Account**:

- **Log In:**
  - The user is asked for their username and password. If correct they "log in" to the application. If not, they are given an error and prompted to give their username and password again.
- **Create An Account:**
  - When creating an account, it checks to make sure the username and email address input are unique. There are no requirements are hashing done to the password, these additions seemed out of the scope of this project.

- **Main Menu:**

The user can choose between **Account**, **Battle**, **Quest Log**, **Marketplace**, **Delete Account** or **Exit**:

- **Account:**
  - In the **Account** menu, the user can:
    - **View Characters:** Prints the character table using the logged in accounts user\_id. They are then asked if they want to **create a character**, **rename a character**, or **return** to the account menu.
    - **View Inventory:** Prints the inventory table using the logged in accounts user\_id
    - **Main Menu:** Returns the user back to the main menu (with choices account, battle, quest log, and exit.)

- **Battle:**
  - Battle function is currently not implemented in this build. The concept of battle is that the user can pick what enemy that want to battle and based on the damage stat of the player and the enemy, the player would either win or lose. If they win they would collect the loot rewarded by the enemy from the **drops** table
- **Quest Log:**
  - The quest log tracks user's quest and if they have completed them. There is currently no quest tracking implemented so all the user can do right now is view their quest.
- **Marketplace:**
  - Accessing the marketplace shows the user what's in the marketplace. Currently there is no interaction the user can do in the marketplace.
- **Delete Account:**
  - The user is asked to confirm their action to make sure they want to delete their account. If so, it goes through tables in the database and deletes the user\_id from the table.
- **Exit:**
  - Exits the application

### Test Queries:

```
SELECT
    user_id,
    username,
    email,
    first_name,
    last_name,
    phone_num
FROM
    user;
```

1	user1	user1@gmail.com	Jacob	Kinchen	985-415-2386
2	user2	user2@gmail.com	Maddie	Jackson	952-153-1936
3	user3	user3@gmail.com	Emily	Shapiro	924-128-1920

4	user4	user4@gmail.com	Avery	Harrell	937-294-5829
---	-------	-----------------	-------	---------	--------------

SELECT

character.name AS Character\_Name,

character.class AS Class,

character.level AS Level

FROM

char\_list

JOIN

character ON char\_list.character\_id = character.character\_id

WHERE

char\_list.user\_id = 1

Character_Name	Class	Level
Barlo	Rogue	8
Bob	Knight	10

```

SELECT
    quest.name AS Quest_Name,
    quest.description AS Description,
    CASE
        WHEN quest_log.is_completed = 1 THEN 'Completed'
        ELSE 'Not Completed' END AS Status
FROM
    quest_log
JOIN
    quest ON quest_log.quest_id = quest.quest_id
WHERE
    quest_log.user_id = 1

```

Quest_Name	Description	Status
Goblin Slayer	Slay 10 Goblins	Completed
Slime Poacher	Slay 20 Slimes	Not Completed
Bully Be Gone!	Slay 1 Bandit	Completed
Sherrif	Slay 30 Bandit	Completed

```

SELECT
    item.name AS Item_Name,
    SUM(contains.amount) AS Amount
FROM
    owns
JOIN
    inventory ON owns.inventory_id = inventory.inventory_id
JOIN
    contains ON inventory.inventory_id = contains.inventory_id
JOIN
    item ON contains.item_id = item.item_id
WHERE
    owns.user_id = 1
GROUP BY
    item.name;

```

Item_Name	Amount
Apple	17
Cloth	10
Goblin Tooth	3
Gold Bar	1
Slimeball	30
Wooden Sword	1

```
SELECT
    user.user_id,
    user.username,
    user.email
FROM
    user
LEFT JOIN
    char_list ON user.user_id = char_list.user_id
WHERE
    char_list.user_id IS NULL;
```

user_id	username	email
4	user4	user4@gmail.com