

Class Diagram and Project Analysis: Equipment Manager in a game, World of Warcraft.

Project analysis :

1. Context and objective :

The program provides a method for managing equipment rewards and player wishes in the game World of Warcraft (WoW). This involves developing an application allowing players to make a wish list for each boss, which the raid leader can consult and assign equipment. Players will also be able to consult and edit their list according to what they have received/wish following the evolution of the game and their equipment.

2. App needs :

- **Player Database** : A secure DB where player can register, login with their credentials, and fill their class, role in raid, role in guild, pseudo in game.
- **Loots & Boss database** : Database with for the bosses, their rewards, and the specs of each rewards.
- **Graphical interface** : A simple and user-friendly graphical interface which allows you to carry out the various actions offered by the app.
- **Player and wishlist management** : The app allows player to register, fill their wishlist and update it.
- **Rewards management**: The app allows the raid leader to manage the rewards for each boss, saying which player recieved which items at which date.

4. Entity analysis :

- **Player (Player)** : Represents a player with their attributes such as username, password, role in raid and guild, class, in game tag.
- **Boss (Boss)** : Represent a boss and his loot table.
- **Wishlist (Wishlist)** : Represent the list of all loot that the player wishes to have.
- **rewards (Loot)** : Represent a reward, it's name, stats, restriction and which boss provides it.
- **Player database (DBPlayer)** : Manage player, we should be able to add, modify, search, and delete them. (only the user can view it's personnal info, the others can only access their in game name, guild and game role.)
- **Boss database (DBBoss)** : manage the boss, players can search, list and view them, admin can modify and add entries.
- **Rewards database (DBLoot)** : Manage the rewards, players can search, list and view them ; admin only can modify and add entries.

5. Key Features :

- Registration and managements of player accounts.
- Creation and management of wishlists for each player.
- View and modify of wishlist by the raid leader and their owner.
- View of available rewards and recording of recieved rewards by the raid leader.

6. Needed technology:

- **Language** : Python
- **Database** : MySQL
- **Graphical Interface** : Tkinter.

7. Stages :

1. Design of the class diagram and general architecture.
 2. Implementation of classes and methods to manage players, bosses and loots.
 3. Setting up the DB to store players, bosses and loots informations.
 4. Development of the GUI.
 5. Security integration, including player authentication and access rights management.
 6. Tests to ensure the good working of the app.
-

Class Diagram :

