

06 December 2020

Server Programming Final Project Documentation

Project Web Application
Project Topic Roleplaying Game Manager
Links Heroku Deployment: <https://rpgmanagerproject.herokuapp.com/>
GitHub Repository: <https://github.com/lanakprojects/rpgmanager>
Project Review: <https://www.youtube.com/>

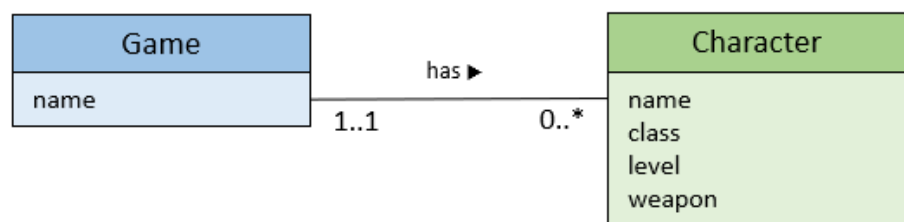
Authentication and Authorizations

Username: Gamemaster
Password: gmPassword
Role: ADMIN

Username: Player
Password: playerPassword
Role: USER

Application Design

1. Database Design 1.1. ER Diagram



1.2. Relational Schema

Game (gameId, gameName)
Char (charId, charName, charClass, charLevel, charWeapon, game)
FK (game) REFERENCES Game (gameId)

1.3. Repository

1.3.1. Entity Type Descriptions

| Entity | Description | Synonyms, aliases | Occurrence |
|-----------|---|-----------------------|--|
| Game | A tabletop or pen-and-paper roleplaying game in which players represent their roles as characters in a fictional setting. | RPG, roleplaying game | A new name of the game can be added only by the owner of the database. The application comes with a pre-set list of the most popular RPGs. |
| Character | A fictional character in an RPG whose actions are controlled directly by the player and progress depends on the course of the game. | char | New characters can be added and edited freely by the players at any time in accordance with the game progression. A character can only be deleted by the Game Master (usually in the case of death of the character as a result of the story progression). |

06 December 2020

1.3.2. Attribute Type Descriptions

Game

| Attribute type | Description | Data type | Value required | Identity attribute or part of it |
|----------------|--------------------------|-----------|----------------|----------------------------------|
| name | The official game title. | Text | + | + |

Character

| Attribute type | Description | Data type | Value required | Identity attribute or part of it |
|----------------|---|-----------|----------------|----------------------------------|
| name | Personal name or nickname of the fictional character. | Text | + | + |
| class | A job or profession, for example, battle or crafting one, that a character specializes in. | Text | + | - |
| level | Progress identificatory for a specific character that can define their skills and abilities, ability power, game difficulty, etc. | Integer | + | - |
| weapon | Main battle equipment or tool utilized by the character. | Text | + | - |

1.3.3. Table Descriptions

Game

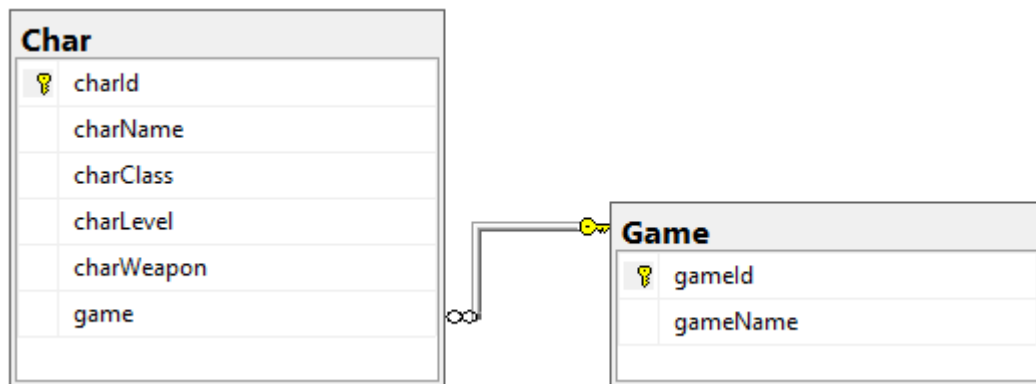
| Column | Description | Data type | Value required | Keys | Special domain |
|----------|-------------------------|-----------|----------------|-----------------------------|-------------------|
| gameId | Unique game identifier. | Long | + | PRIMARY KEY (surrogate key) | Autonumber column |
| gameName | Title of the game. | String | + | - | - |

Char

| Column | Description | Data type | Value required | Keys | Special domain |
|------------|---|-----------|----------------|-----------------------------|-------------------|
| charId | Unique character identifier. | Long | + | PRIMARY KEY (surrogate key) | Autonumber column |
| charName | Name of the character. | String | + | - | |
| charClass | Assigned class of the character. | String | - | - | |
| charLevel | Current level of the character. | int | - | - | |
| charWeapon | Main weapon currently equipped by the character. | String | - | - | |
| game | Unique identifier and title of the game the character belongs to. | Game | - | FOREIGN KEY => Game | Valid game name. |

06 December 2020

1.4. Database Diagram



2. Application Functionality Description

| Function | Role | Description |
|----------|-------------|---|
| List | USER, ADMIN | Allows to view all the characters from the database and their attributes. |
| Add | USER, ADMIN | Adds new character to the database. |
| Edit | USER, ADMIN | Allows to change the attributes of the character. |
| Delete | ADMIN | Deletes a character from the database. |
| Sort | USER, ADMIN | Sort characters by name, class, level, weapon, or game name. |

3. Testing

3.1. Character Repository Test

Tests CharRepository interface functionality including finding the character by the name, creating new character, and deleting a character.

3.2. Game Repository Test

Tests Game Repository interface find by the name method.

3.3. User Repository Test

Tests UserRepository interface find by the username method.