# Dungeons and Dragons Management

# Problem Statement

**Rose-Hulman Institute of Technology - CSSE 333**

Alec Goodrich

Andrew Orians

Cade Parkhurst

# Table of Contents



# Version Information

| Version | Date | Comments |
|---|---|---|
| 1.0 | 1/8/21 | Initial Draft |

# 1. Executive Summary

This document's purpose is to describe the problem that our project will solve. For better understanding an Entity Relation (ER) diagram is provided. This document is the first of multiple documents that describes the problem. A high-level problem summary, a detailed problem statement, and some information about the stakeholders are included in this document.

Dungeons and Dragons is one of most popular role-playing games that can be played [1]. One problem players have always faced, is how to effectively store the information about their characters. Additionally, players constantly have to look up information about their characters due to the large amount of information. This can hinder and slow down a person's experience with Dungeons and Dragons. We propose a software solution designed to be efficient, simple, and informative tracking of player characters. This system will give players the ability to know more about their characters and manage their characters easier.

# 2. Introduction

This document is the first document describing our (Project name) management system. For greater understanding of the system an ER diagram is included. Following this document will be a relational schema, a security analysis, some periodic reports, and a final presentation. This document will give a brief overview of the proposed system and its requirements. The security and data analysis documents will go into more detail. The relational schema will describe the database and foreign key constraints, based upon the ER diagram. The final presentation will demonstrate the completed system, as well as describing the process we used in creating the system.

# 3. High Level Problem Summary

## 3.1 Elevator Statement

We are designing a business management system for Dungeons and Dragons. Dungeon and Dragon players have always had troubles with managing and understanding their characters. Due to this, player's experience with Dungeon and Dragons is worsened. We propose a software solution designed to be efficient, simple, and informative for character management.

## 3.2 Primary Success Criteria

Our primary goal is to provide Dungeons and Dragons players with a way to manage their characters and access information about those characters. There is a lot of information to keep track of when managing characters and accessing character information. The project's success depends upon having a system that manages and stores information about characters based on the Player's Handbook.

## 3.3 Scope

### *3.3.1 Within Scope*

1. Classes
2. Sub-Classes
3. Spells
4. Items
5. Races
6. Feats
7. Skills
8. Saving Throws
9. Weapons
10. Armor
11. Background
12. Homebrew

### *3.3.2 Outside of Scope*

1. Information from sources other than the Player's Handbook
2. Monsters
3. Transportation
4. Game Rules

# 4. Detailed Problem Statement

## 4.1 Key Stakeholders

| **Name** | ***Role*** |
|---|---|
| Amanda Stouder | Project Advisor |
| Andrew Orians | Project Team |

| | |
|---|---|
| Alec Goodrich | Project Team |
| Cade Parkhurst | Project Team |

### 4.2 Function

1. Ability to track the current stats of a character
    a. HP
    b. Max HP
    c. Spell Slots
    d. Armor Class
    e. Ability Scores
    f. Saving Throws
    g. Skills
    h. Speed
    i. Initiative Bonus
2. Several types of player equipment will be tracked
    a. Money
    b. Weapons
    c. Armor
    d. Tools
    e. Useable items
3. Each character has its own set of stats, equipment, and traits
4. Ability to track the various traits a character has
    a. Class traits
    b. Race traits
    c. Background traits
5. Ability to look up the meaning of things on the character sheet
    a. Spells
    b. Traits
    c. Background
    d. Class
    e. Items
6. Ability for users to login
7. Multiple characters can be assigned to one user

## 5. References

[1] Dungeons and Dragons: https://en.wikipedia.org/wiki/Dungeons_%26_Dragons#:~:text=Advanced%20Dungeons%20%26%20Dragons%20was%20ranked,Hall%20of%20Fame%20in%202017.

## 6. Appendix

Items(ItemID, Cost, Weight, Name)

Weapon(ItemID, Damage, Properties, WeaponTypeName)

WeaponType(Name)

Armor(ItemID, Name, BaseAC, AddDex?, StrengthMin, DisAdvantageOnStealth?, ArmorWeightName)

ArmorWeight(Name)

Spell(SpellID, CastingTime, Range, V?, S?, M?, Duration, Concentration?)

Class(ClassID, HitDice, NumProfieciencies)

GivesClassFeature(ClassFeatureID, ClassID, GainedAtLevel)

Starts_With(ClassID, ItemID, Quantity)

SubClass(SubclassID, ClassID)

GivesSubclassFeature(SubclassFeatureID, SubClassID, GainedAtLevel)

Chose_Subclass(CharacterID, SubClassID)

Language(Name)

Knows_Language_From_Race(RaceName, LanguageName)

Knows_Language(CharacterID, LanguageName)

Background(Name, Feature, NumLanguagesGained)

Background_Skill_Prof(BackgroundName, SkillName)

Background_Tool_Prof(BackgroundName, ItemID)

Race(Name, AbilityScore, NumIncreasedBy, SubRace)

Race_Weapon_Prof(RaceName, WeaponID)

Race_Tool_Prof(RaceName, ItemID)

GivesRaceFeature(FeatureID, RaceName)

Character(CharacterID, Str, Dex, Int, Wis, Char, Con, Alignment, HP, MaxHP)

Has_Levels_In(CharacterID, ClassID, NumLevels)

Is_Owned_By(ItemID, CharacterID, Quantity)

Has_ArmorWeight_Prof(ClassID, ArmorWeightName)

Has_WeaponType_Prof(ClassID, WeaponTypeName)

Has_Weapon_Prof(ClassID, WeaponID)

Offers_Skill_Proficiency(ClassID, SkillName)

Chose_Proficiency(CharacterID, SkillName)

Has_SavingThrow_Prof(ClassID, SavingThrowName)

KnowsSpell(CharacterID, SpellID)

Can_Learn_Spell(ClassID, SpellID)

Skill(Name)

SavingThrow(Name)

Trait(TraitID, Description, Type)

**Foreign Keys**

Weapon(ItemID) -> Item(ItemID)

Weapon(WeaponType) -> WeaponType(Name)

Armor(ItemID) -> Item(ItemID)

Armor(ArmorWeightName) -> ArmorWeight(Name)

GivesClassFeature(ClassID) -> Class(ClassID)

GivesClassFeature(ClassFeatureID) -> Trait(TraitID)

Starts_With(ClassID) -> Class(ClassID)

Subclass(ClassID) -> Class(ClassID)

GivesSubclassFeature(SubclassFeatureID) -> Trait(TraitID)

GivesSubclassFeature(SubclassID) -> Subclass(SubclassID)

Chose_Subclass(CharacterID) -> Character(CharacterID)

Chose_Subclass(SubclassID) -> Subclass(SubclassID)

Knows_Language_From_Race(RaceName) -> Race(Name)

Knows_Language_From_Race(LanguageName) -> Language(Name)

Knows_Language(CharacterID) -> Character(CharacterID)

Knows_Language(LanguageName) -> Language(Name)

Background_Skill_Prof(BackgroundName) -> Background(Name)

Background_Skill_Prof(SkillName) -> Skill(Name)

Background_Tool_Prof(BackgroundName) -> Background(Name)

Background_Tool_Prof(ItemID) -> Item(ItemID)

Race(Subrace) -> Race(Name)

Race_Weapon_Prof(RaceName) -> Race(Name)

Race_Weapon_Prof(WeaponID) -> Weapon(ItemID)

Race_Tool_Prof(RaceName) -> Race(Name)

Race_Tool_Prof(ItemID) -> Item(ItemID)

GivesRaceFeature(FeatureID) -> Trait(TraitID)

GivesRaceFeature(RaceName) -> Race(Name)

Has_Levels_In(CharacterID) -> Character(CharacterID)

Has_Levels_In(ClassID) -> Class(ClassID)

Is_Owned_By(ItemID) -> Item(ItemID)

Is_Owned_By(CharacterID) -> Character(CharacterID)

Has_ArmorWeight_Prof(ClassID) ->  Class(ClassID)

Hass_ArmorWeight_Prof(ArmorWeightName) -> ArmorWeight(Name)

Has_WeaponType_Prof(ClassID) -> Class(ClassID)

Has_WeaponType_Prof(WeaponTypeName) -> WeaponType(Name)

Has_Weapon_Prof(ClassID) -> Class(ClassID)

Has_Weapon_Prof(WeaponID) -> Weapon(ItemID)

Offers_Skill_Proficiency(ClassID) -> Class(ClassID)

Offers_Skill_Proficiency(SkillName) -> Skill(Name)

Chose_Proficiency(CharacterID) -> Character(CharacterID)

Chose_Proficiency(SkillName) -> Skill(Name)

Has_SavingThrow_Prof(ClassID) -> Class(ClassID)

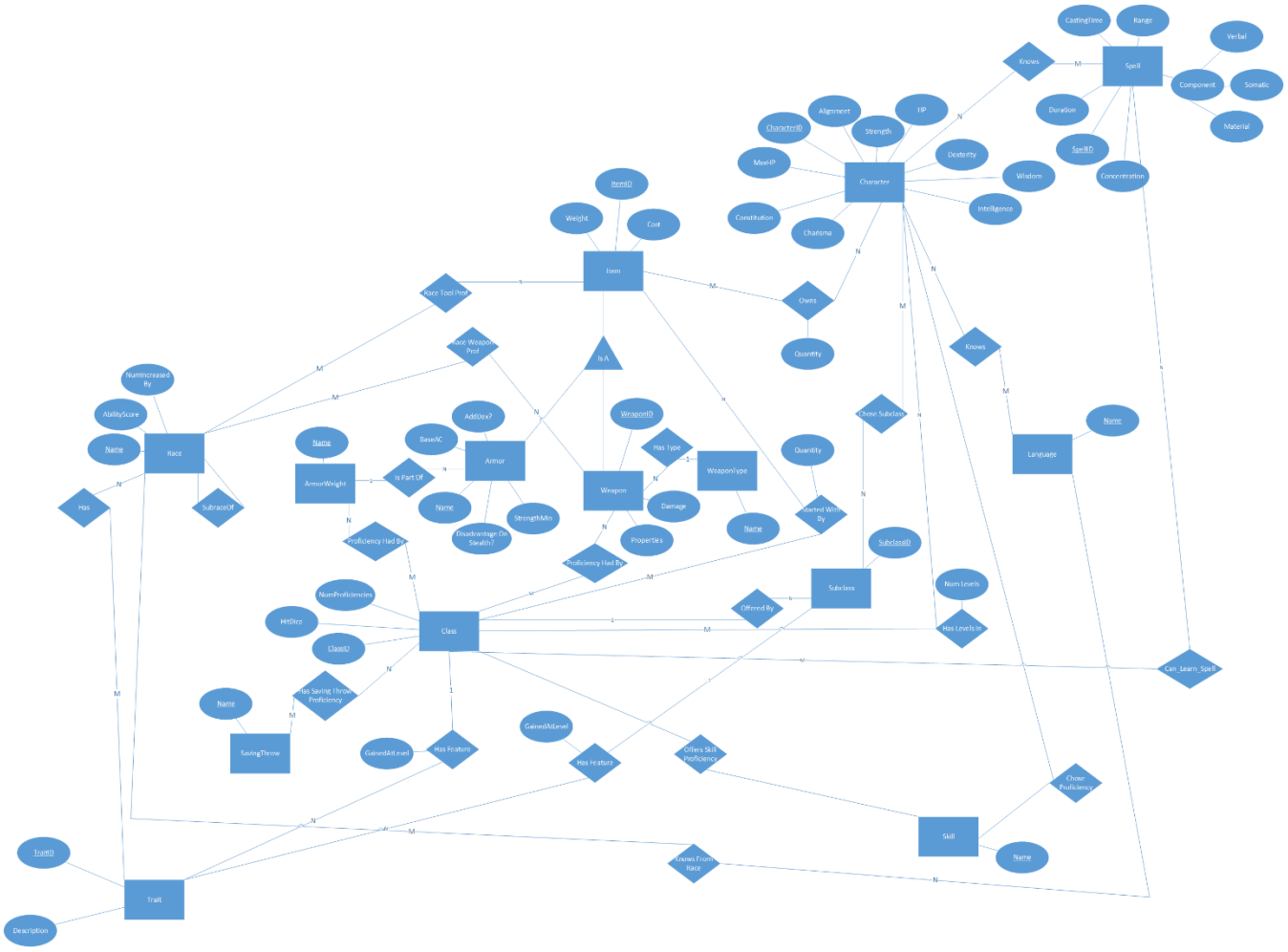
Has_SavingThrow_Prof(SavingThrowName) -> SavingThrow(Name)

KnowsSpell(CharacterID) -> Character(CharacterID)

KnowsSpell(SpellID) -> Spell(SpellID)

Can_Learn_Spell(ClassID) -> Class(ClassID)

Can_Learn_Spell(SpellID) -> Spell(SpellID)




# 7. Index



# 8. Glossary

Entity relationship (ER) diagram - An abstract way of representing the layout of a database.

Dungeons and Dragons – A popular tabletop role-playing game

Class – Type a character you are playing  ex: Cleric or Wizard

Subclass - The specific type of class you are, (Necromancer vs plain wizard)

Proficiency – When your character is trained in/good at something