Project Proposal

Top level Description:

Data will be gathered from an Excel document listing player characters from the popular Role Playing game Dungeons and Dragons. This information will include attributes such as the player character's class, level, stats, and personality. We hope to gather this data from a Google form we will distribute to the Northeastern Roleplaying Society and potentially online Dungeon and Dragon forums. We will organize this data into tables for players, personalities, classes and races.

Our code will determine the average statistics for players in general, each class, each race, and each level. They will also determine common aspects of player personalities by searching for common words in player personality descriptions. Users will be able to look up characters based on character names, and compare character statistics and personalities to those of all other characters, characters of their class, and characters of their race. Users will be able to add new characters, remove characters if they have administrator privileges, and update characters if they are the owner of that character. Ownership will be determined by the MySQL user name system.

We plan to have a character view, which will show data on a on a select number of characters; a race view which will display all information on a select number of races; and a class view which will display information on a select number of classes.

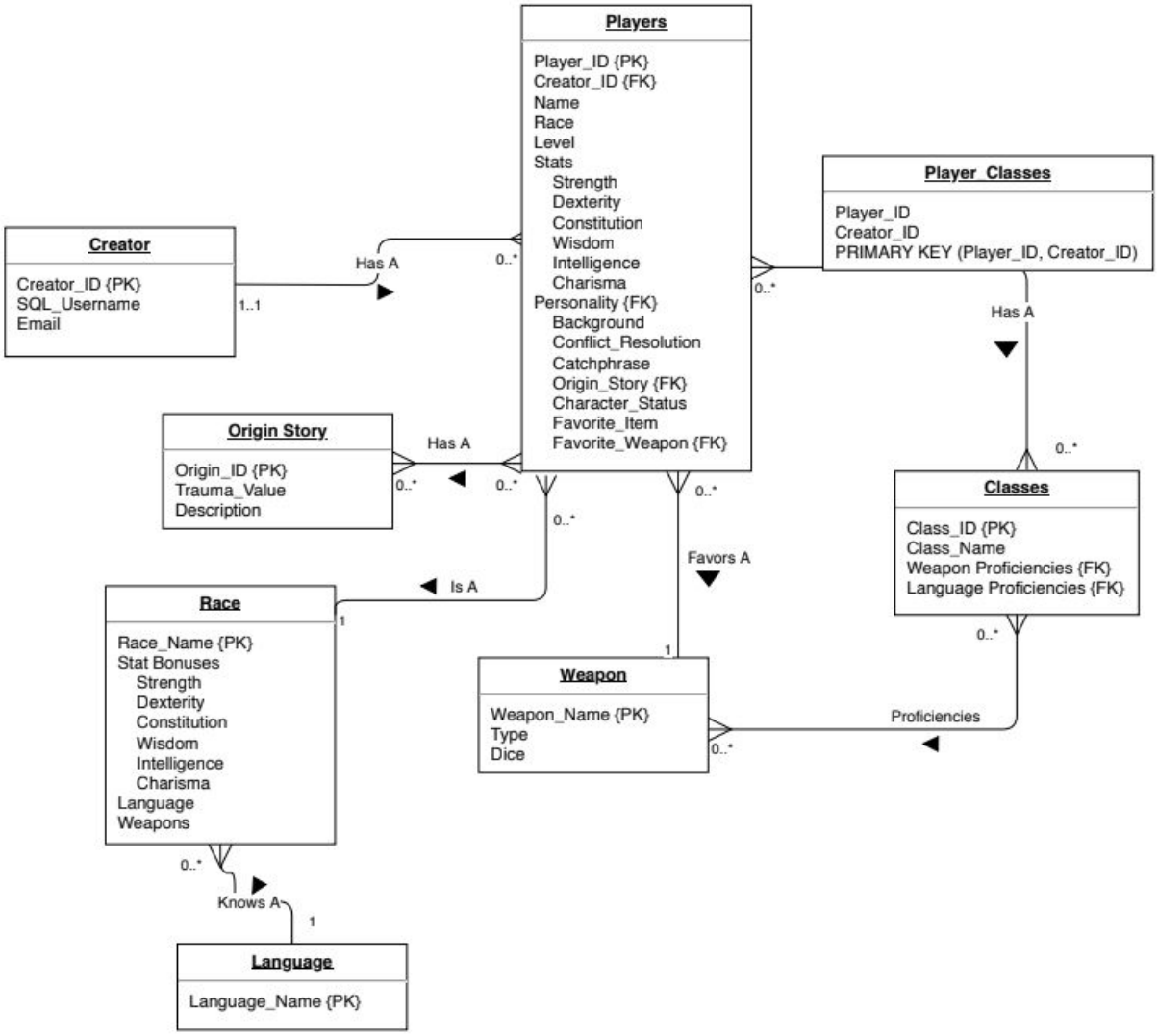
Software:

We plan is to use SQL storage for our relational database, and the Java language to connect to it. We will use the Java GUI library for our frontend view, displaying information and allowing the user to put in input into the program. To gather data for the database, we plan to use Google forms, whose data we plan to download into csv form. For developing the project, we will use our own laptops to write code, and Github to help us merge our code together.

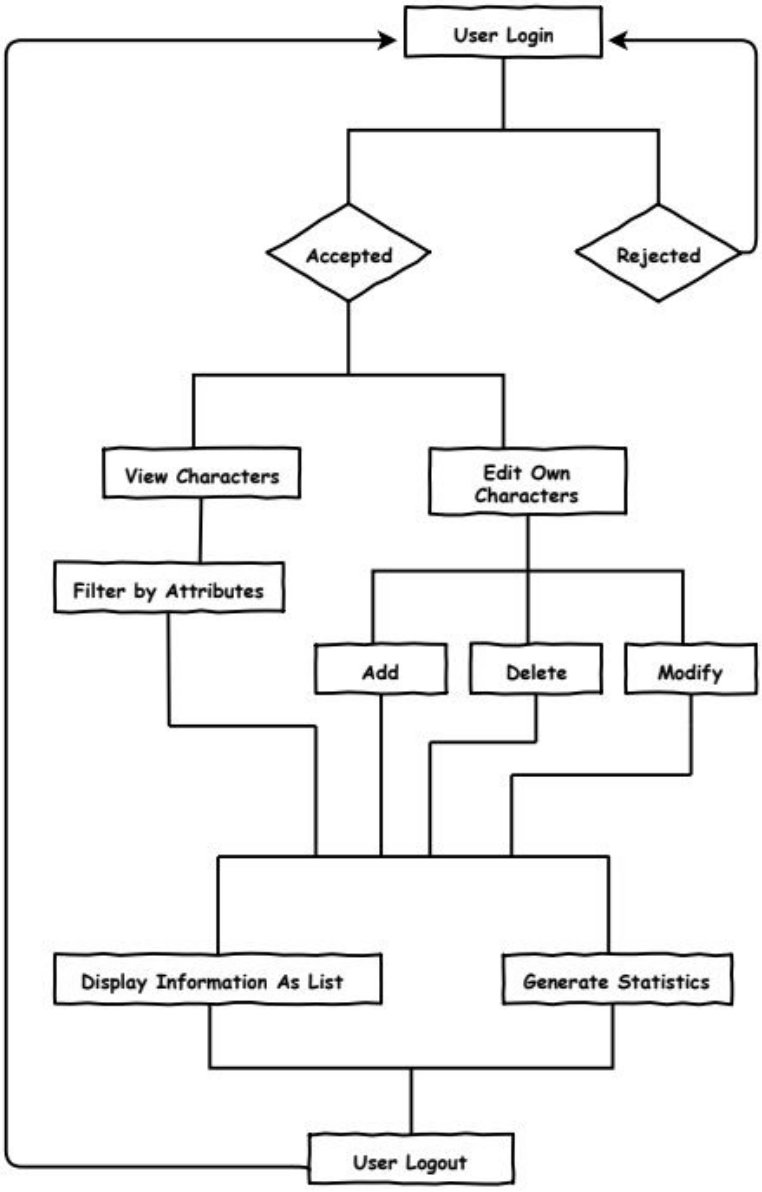
Interest:

Both of us are players of dungeon and dragons, and are very curious as to the statistics of characters that other players generally create. We find worthwhile to compare this to our own characters, and to see whether certain aspects of the characters, such as class, have any trends of their own.

UML Diagram


Players
Player_ID {PK}
Creator_ID {FK}
Name
Race
Level
Stats
Strength
Dexterity
Constitution
Wisdom
Intelligence
Charisma
Personality {FK}
Background
Conflict_Resolution
Catchphrase
Origin_Story {FK}
Character_Status
Favorite_Item
Favorite_Weapon {FK}
Creator
Creator_ID {PK}
SQL_Username
Email
Has A
1..1
0..*
Player_Classes
Player_ID
Creator_ID
PRIMARY KEY (Player_ID, Creator_ID)
0..*
Has A
0..*
Origin Story
Origin_ID {PK}
Trauma_Value
Description
Has A
0..*
0..*
0..*
0..*
Favors A
Classes
Class_ID {PK}
Class_Name
Weapon Proficiencies {FK}
Language Proficiencies {FK}
Is A
1
Race
Race_Name {PK}
Stat Bonuses
Strength
Dexterity
Constitution
Wisdom
Intelligence
Charisma
Language
Weapons
1
Weapon
Weapon_Name {PK}
Type
Dice
0..*
Proficiencies
0..*
0..*
Knows A
1
Language
Language_Name {PK}

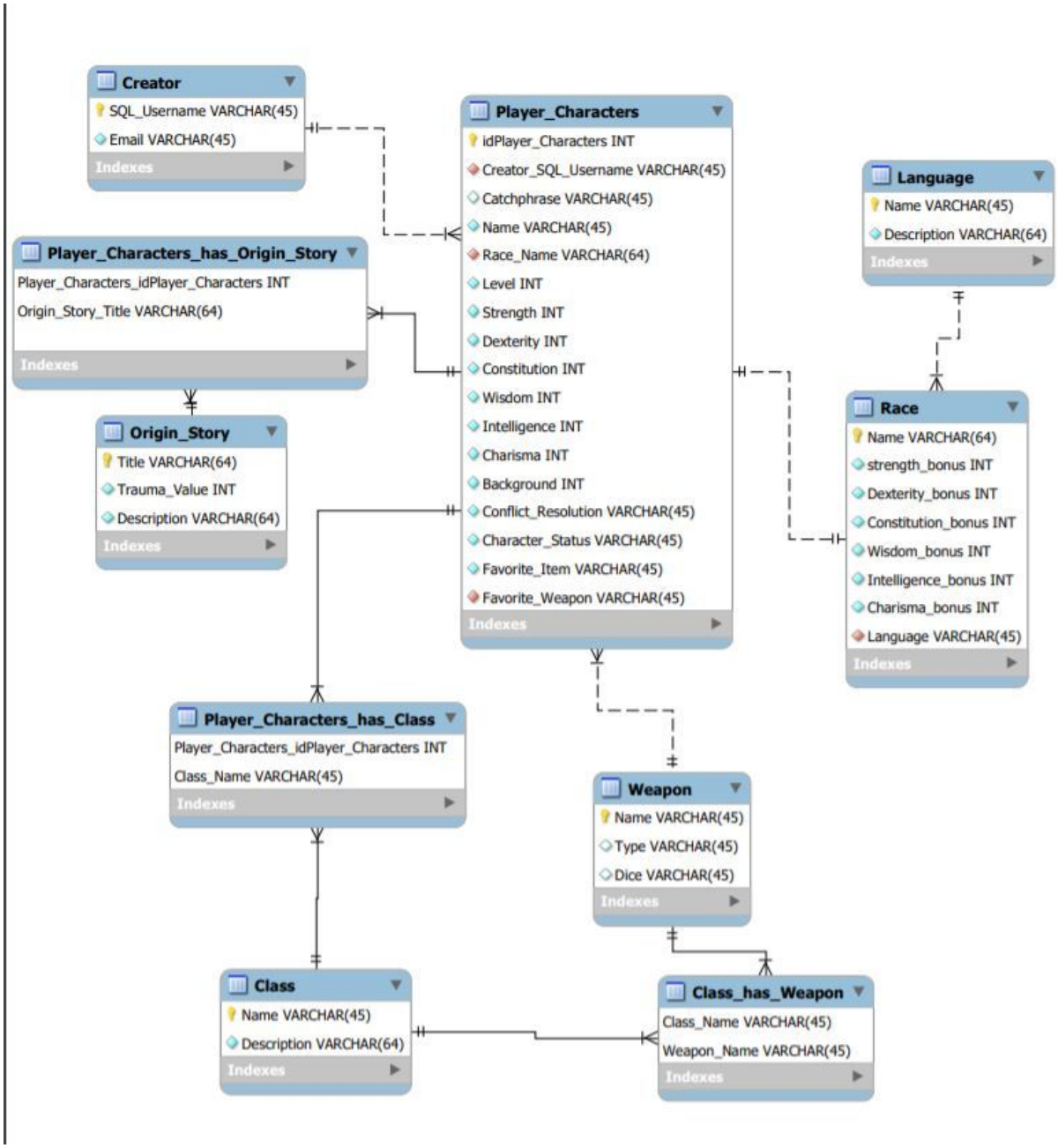
Flow Chart



1. The user will log into their account with a username and password. They will either be accepted or rejected
    a. If rejected, they will be sent back to the login page to try again
    b. If accepted, continue
2. Once logged in, the player can choose to view the database of characters or edit their own characters
    a. While editing characters, they will have a change to add characters, delete characters, or modify existing characters

    b. While viewing characters, they may filter by attributes, such as selecting specific classes, races, and weapons and seeing which characters have these characteristics
3. After editing or selecting filters, player may :
    a. either view a table of all characters with selected traits
    b. generate overall statistics of the characters satisfying the filter requirements.
4. User may then log out of the page, which will bring them back to the login

EER Diagram


Creator
SQL_Username VARCHAR(45)
Email VARCHAR(45)
Indexes
Player_Characters
idPlayer_Characters INT
Creator_SQL_Username VARCHAR(45)
Catchphrase VARCHAR(45)
Name VARCHAR(45)
Race_Name VARCHAR(64)
Level INT
Strength INT
Dexterity INT
Constitution INT
Wisdom INT
Intelligence INT
Charisma INT
Background INT
Conflict_Resolution VARCHAR(45)
Character_Status VARCHAR(45)
Favorite_Item VARCHAR(45)
Favorite_Weapon VARCHAR(45)
Indexes
Language
Name VARCHAR(45)
Description VARCHAR(64)
Indexes
Player_Characters_has_Origin_Story
Player_Characters_idPlayer_Characters INT
Origin_Story_Title VARCHAR(64)
Indexes
Origin_Story
Title VARCHAR(64)
Trauma_Value INT
Description VARCHAR(64)
Indexes
Race
Name VARCHAR(64)
strength_bonus INT
Dexterity_bonus INT
Constitution_bonus INT
Wisdom_bonus INT
Intelligence_bonus INT
Charisma_bonus INT
Language VARCHAR(45)
Indexes
Player_Characters_has_Class
Player_Characters_idPlayer_Characters INT
Class_Name VARCHAR(45)
Indexes
Weapon
Name VARCHAR(45)
Type VARCHAR(45)
Dice VARCHAR(45)
Indexes
Class
Name VARCHAR(45)
Description VARCHAR(64)
Indexes
Class_has_Weapon
Class_Name VARCHAR(45)
Weapon_Name VARCHAR(45)
Indexes

Technical Specifications

We plan is to use SQL storage for our relational database, and the Java language to connect to it. We will use the Java GUI library for our frontend view, displaying information and allowing the user to put in input into the program. To gather data for the database, we plan to use Google forms, whose data we plan to download into csv form. For developing the project, we will use our own laptops to write code, and Github to help us merge our code together.