Terraria equipment databAse

IT&C 350 database design project

Winter 2024

Sam Swindler

Ethan Beere

Matthew Gregg

Spencer Baird

Table of Contents

Project Overview 3

Project Objective Statement 3

Project Stakeholders 3

App Requirements 4

Functional Requirements 4

Non-Functional requirements 4

Database Requirements 5

ER Diagram Images 5

Schema Diagram 5

Business Rules 6

Relational Schema Types 6

Database Documentation 7

GitHub Access 7

Accessing the Database 7

Data needed 7

# Project Overview

## Project objective Statement

Create an equipment preparation system for Terraria players that enables users to discover and create the best combination of equipment for their current in-game situation.

## Project Stakeholders

Who are the Stakeholders?

* Terraria players.
* Our creation team.

Who is going to be maintaining the site?

* We, the aforementioned four horsemen of this project, are the maintainers of the site.

A shortened interview with Mr. Sam, a Terraria expert.

Q: Why do you like Terraria?

A: There’s lots of bosses and lot of fun equipment. You can kill things with yo-yo's!

Q: What is the hardest thing about Terraria?

A: Acquiring stuff and getting ready for fights. Need to understand how to get ready.

Q: What would you envision an app of this type could do for you?

A: It could help me understand my options at where I am in the game.

Q: Limitations of current systems?

A: Terraria wiki exists, but you have to browse the wiki. Stats aren’t shown in a concise form.

Q: What features would you like to see?

A: Understand boss difficulties, and crafting trees.

# App Requirements

## Functional Requirements

Home Page Functionality

* Users can see a list of all items available for them to acquire and use under the condition that they have logged in and created a character profile.
* The list of available items changes based on the information provided by users about their Terraria Character.
* Users can view individual items with statistics about each item.
* Users can select an item to add to their list of equipped items.

Character Page Functionality

* Users can record what bosses their character has defeated so far.
* Users can record whether they have entered hard-mode or not.
* Users can record what biomes/regions they have discovered in their world so far.
* Users can see the current character statistics with the items they have equipped.

User page Functionality

* Users can create and select characters.
* Users can change their password.

## non-Functional Requirements

Security

* Website protects against XSS.
* Website inputs will protect against SQL injections.
* Fail2ban (wish list)

Availability

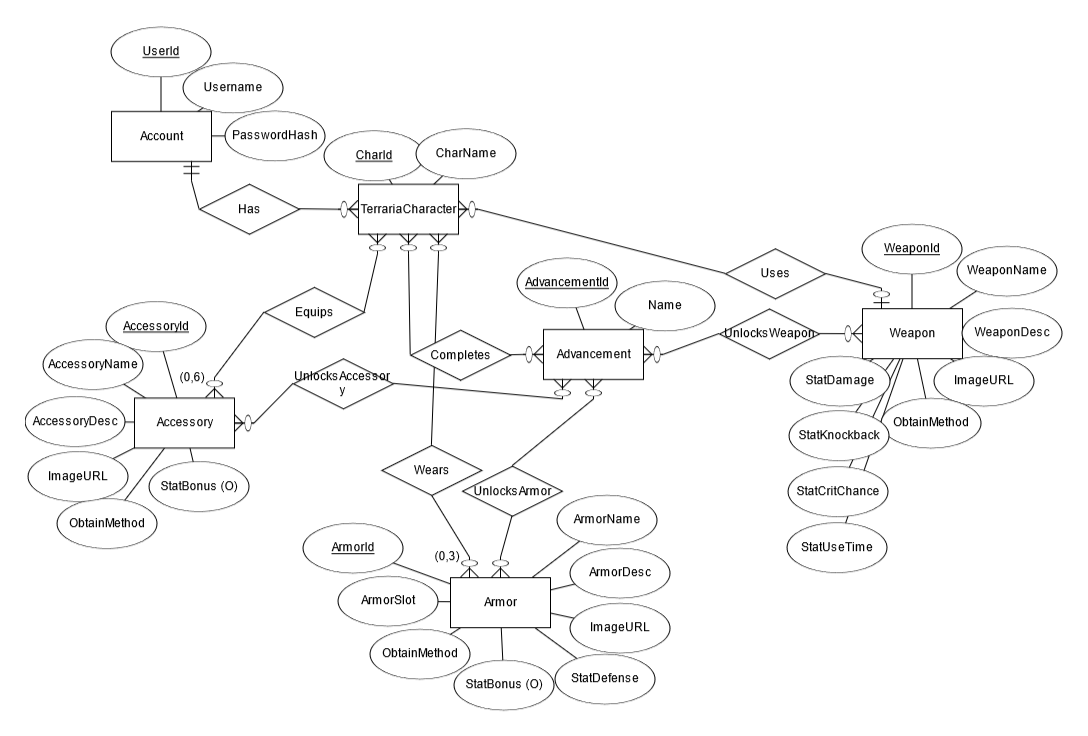
* 95% uptime and availability.

Usability

* Mobile/Desktop application. (wish list)
* Dark mode.

# Database Requirements

## ER Diagram Images



## Schema Diagram

## Business Rules

* Users can create up to 25 characters.
* Newly created characters have no items assigned to them.
* Characters can equip, at maximum, 1 weapon, 3 armor pieces (head, chest plate, boots), and 6 pieces of equipment.
* Users can have zero or one characters selected for viewing relevant and equipped items.
* The list of items cannot be changed from the front end and will be managed directly in the database by site administrators when needed.
* Usernames can only contain letters and numbers (no special characters).
* Attempts to equip more items than is permitted will not replace prior equipped items (operation will produce an error instead).
* If a user has no character selected, they are unable to equip any items.
* Users can have zero or more advancements selected when filtering data output.

## Relational Schema Types

**Account:**

Username (varchar)

PasswordHash (varchar)

UserID (int)

**Character:**

CharID (int)

CharName (varchar)

**Weapon:**

WeaponID (int)

WeaponName (varchar)

WeaponsDesc (varchar)

ImageURL (varchar)

ObtainMethod (varchar)

StatAttack (varchar)

**Accessory:**

AccessoryID (int)

AccesoryName (varchar)

AccessoryDesc (varchar)

StatBonus (varchar)

ObtainMethod (varchar)

ImageURL (varchar)

**Armor:**

ArmorID (int)

ArmorName (varchar)

ArmorDesc (varchar)

ImageURL (varchar)

StatDefense (int)

ObtainMethod (varchar)

ArmorSlot (int)

StatBonus (varchar)

**Advancement:**

Name (varchar)

AdvancementID (int)

# Database Documentation

## Github Access

Scripts for creating, populating, and clearing the database are found on the GitHub repository under the Milestone 3 folder. The GitHub repository is accessible at <https://github.com/suspenceb/TerrariaCraftingRepo>.

## Accessing the Database

1. Connect to the IT&C VPN
2. Open a command prompt and SSH into the server with:
3. ssh username@172.16.32.12
4. Navigate to our Docker folder in /docker.
5. Turn on the Database and PHPMyAdmin with docker compose up -d (It can be turned off with docker compose down) (Information about the current images can be seen with docker ps)
6. Connect to PHPMyAdmin with http://172.16.32.12:8000 in your browser.

Note: For convenience, the domain db.beerefamily.org has been registered to point to 172.16.32.12 and can be used in place of the IP address.

## “Data for pages or something” (Jordan)

The following screenshots will display what data is needed in each screen taken from a mermaid diagram. All view statements are included on the github.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A blue sign with white text

Description automatically generated

# API Documentation

# Front-End Documentation

# Appendix 1: Low-fidelity paper prototypes

|  |
| --- |
| Login Screen |
| Account controls screen |
| Character Information Screen |
| Item Details Screen |
| Home page showing available items |

# Appendix 2: High-Fidelity Paper Prototypes

[images]