Forge Frontier

CS 307
Design Document

Team #19:

- Joseph Miller
- Meet Patel
- Bhavesh Pareek
- Isaac Fleetwood
- Aditya Behrani

Index

Purpose

Minecraft is first and foremost an experience centered around creativity and exploration, but in order to progress significant time must be invested. Many others have attempted to create servers in minecraft which try to provide an idle progression option but in doing so, either offer no enjoyable in-game challenges or lose sight of their original goal and render participation in the idle system meaningless.

We strive to strike the balance between the practical idling progression system accessible in-game and via our website, and the gearing progression system through mining, fighting enemies, and overcoming challenging bosses. By reaching this goal, we hope to serve both players who want to spend hours upon hours getting lost in their character, as well as those who just don't have the freetime, but also want to participate and progress in MMO gameplay in a minecraft server.

Functional Requirements:

1. User Idle Progression:

a. I want a personal island to house generators, builds, chests, etc.
b. Tycoon style item generators - custom upgrades, variety of items, etc.
c. Web integration of generators - collect & view generators
d. I want to be able to passively gain in-game currency through tycoons

2. User Active Progression:

a. I want to progress through mining, farming, fighting monsters and bosses actively.
b. I want to be able to collect materials and items from the above activities.
c. I want to be able to collect and create enhancements that I can apply to my gear.

d. I want to be able to buy and sell items on a global market for prices I set.
e. I want at least 2 unique boss fights that are enjoyable and implement mechanics beyond the scope of base minecraft.

3. User Website Interface:

a. I want to be able to view mine and others' generators and progression on the website..
b. I want to be able to view mine and others' items and stats on the website
c. I want the website to give me a way to link to my Minecraft account.
d. I want to be able to see the market on the website and the cheapest buy & sell prices.
e. I want to be able to see market listings on the website.

4. User Security

a. I want protection from duping exploits on gear.
b. I want the collection of my tycoon generators to be consistent; I want the databases which store my progression to be consistent.
c. I want the link between my Minecraft and Website to be protected via OAuth 2.0
d. I want the game to be balanced, and for this, I want it so that gear is consistent among players.

5. User Nice to Have:

a. I want particle effects on my gear in-game.
b. I want to see a leaderboard on the website to see who is winning.
c. I want to have a variety of NPCs to interact with

Non Functional Requirements

1. **Website**
    a. The website should be responsive and dynamic
    b. Content should ideally refresh automatically instead of requiring a user to press f5
    c. Mobile and desktop sites should be the same codebase but fluently adapt to resizes
    d. User authentication should use secure json web tokens and handle authentication success and errors well
    e. The server should have a reasonable rate limiting setup
    f. User data should be stored in a structured database, ideally managed with automatic backups handled for us
    g. If self hosted, database should have durability measures: backups, health checks
2. **MC Server**
    a. There must be a consistent flow of game data to the SQL server without lag
    b. Plugins must be optimal and created with efficiency in mind; the server should be able to perform at 16+ TPS (Minecraft ticks per second)
    c. Relevant plugins (Gear) should have an interface for other plugins to access and utilize for game logic.

d. Configuration files or database entries must be present for developed plugins where possible; in our case, likely with tycoon generators and gear.
e. Tests must be in place to verify integrity of gear, generators, and monsters; this will be done by checking the properties and locations and ascertaining they are within viable bounds.
f. Player information must be kept synchronized whenever possible; especially with collection of tycoon generators.

Design Outline

Overall Construction

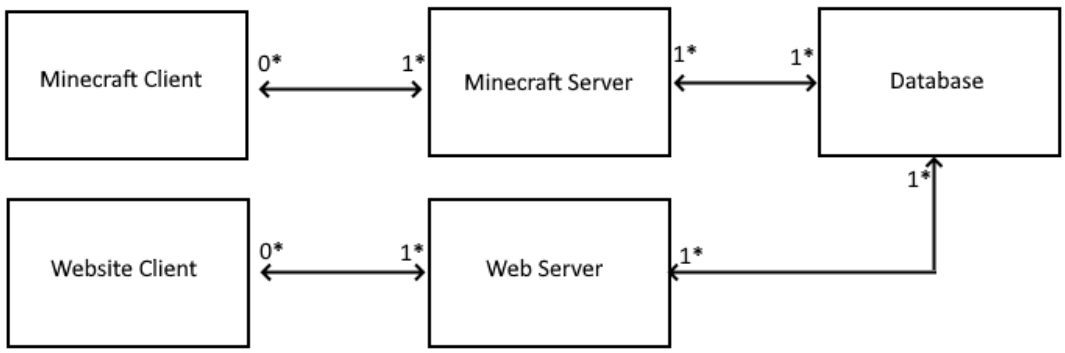
Minecraft Client: Connects to MC server through the regular Minecraft protocol. We do not have any modifications.

Minecraft Server: Spigot API and NMS (Net Minecraft Server) integrations are utilized to modify server side behaviors that we send to the MC client. Plugins (utilizing Spigot API and NMS integrations) will be run on the MC server, and will communicate with the Database. We try to *avoid* connecting to the web server as it is simpler to update the database whenever necessary.

Plugin data sent from MC server to database:

- Tycoon Generator status
- Player stats and inventory
- Market information
- Gear information

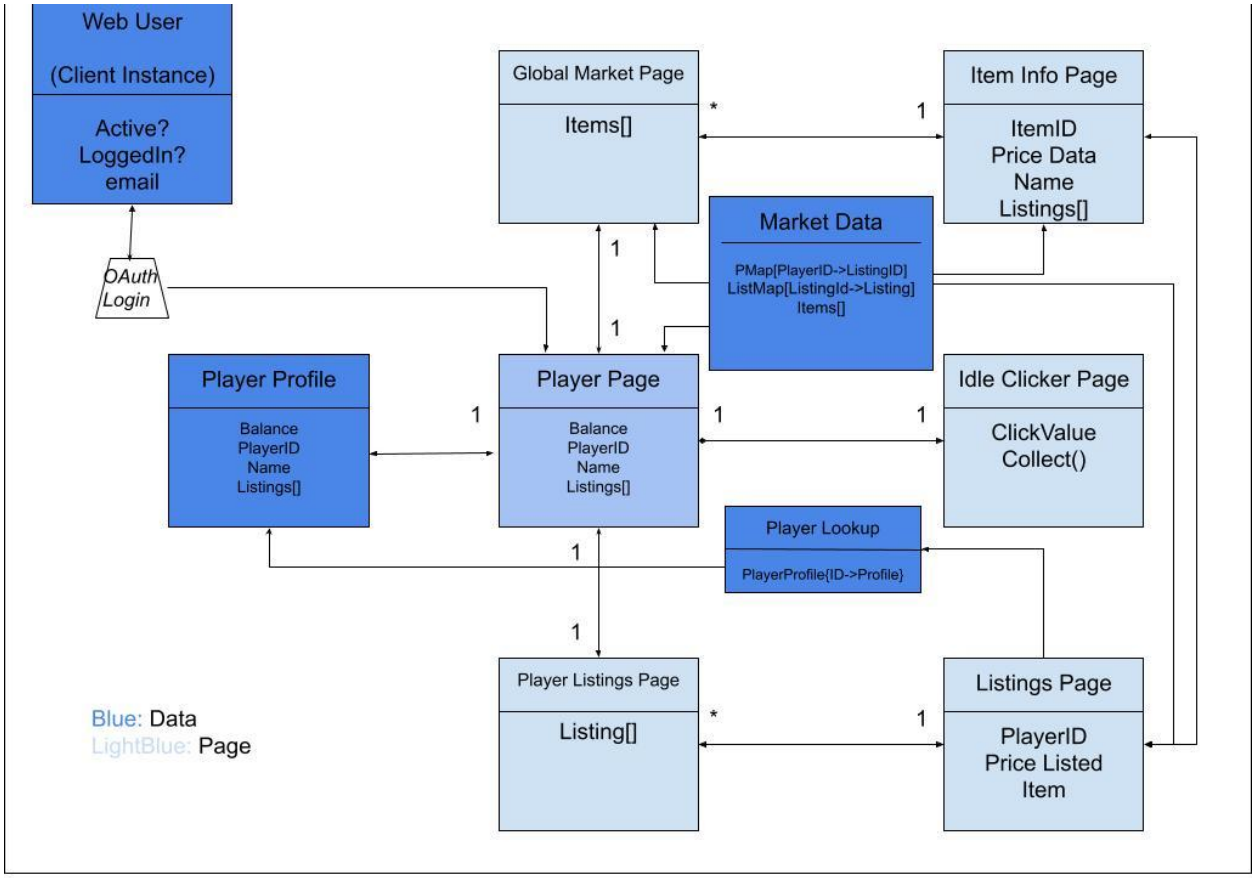
Database: Primary storage of Web accessible features, player data, and game constants. Ex. Generator Status, Player Stats, Market Info, Gear Info.



Website Design

Web Server: Mostly utilized to pull database values for web client view. This includes, but is not limited to, generator status, player stats, market information, gear information.

Web Client: Website interface for viewing game data.



Design Issues

Functional Issues:

1. How will users link their Minecraft account with their account on our website?
    a. Option 1: Link their Microsoft account with their account on our website
    b. Option 2: Use a generated code received from in-game to connect their account

    Choice: Option 2

    Justification: While linking our account with their Microsoft account would allow us more access to their account information, and ensure that the user linking the account has access to their account information, it is unnecessary and would be more of a hassle for the user. By providing a code which the user can then paste into our website (while logged in with their account), the process of linking their account would be incredibly simple and the user would be directed straight to the correct page on the website via a link.

2. Should users be able to collect from their generators on the website?
    a. Option 1: Yes
    b. Option 2: No

    Choice: Option 1

    Justification: Since the main purpose of the website is to allow interaction with the game while not on your computer, it is integral that the user should be able to collect their accrued wealth through the website.

3. Should users be able to trade on the market while on the website?
    a. Option 1: The user can only view the market

b. Option 2: The user can view and interact with the market

Choice: Option 1

Justification: While this feature would be useful for users, allowing them to sell and buy items from the website, its existence would be more harmful than convenient. By not allowing market interaction on the website we would reduce potential botting, by requiring a minecraft account to be used it will increase the difficulty of potential botting.

4. How will user's increase their power?
    a. Option 1: Through buying more powerful gear
    b. Option 2: Through receiving more powerful gear through mob and boss drops
    c. Option 3: By upgrading the gear they already own
    d. Option 4: All of the above options

Choice: Option 4

Justification: Each option 1-3 is a very viable strategy to allow a user to gain more powerful equipment, however by only implementing a very linear path of progression, it is likely to become more boring overtime. By providing multiple paths of progression, the user is encouraged to participate in multiple gameplay activities resulting in a more active user-base.

5. How will user's access the marketplace?
    a. Option 1: Through right-clicking an item
    b. Option 2: By traversing to a marketplace area in the server

Choice: Option 2

Justification: While allowing a user to simply right-click an item to have access to a shop is very convenient for the player, providing a market-place area in game is ultimately more beneficial for the players. This approach essentially creates an in-game area where players can congregate and helps make the game feel more alive. If all players could simply access all features of the server while positioned at their island, there would be little instances of players seeing one another on the server, which would decrease the feeling that they are participating in a shared world. A physical market-place area would help prevent this problem from occurring.

Non Functional Issues

1. How do we handle users with unsupported browsers / no js?
    a. Option 1: Block site access entirely
    b. Option 2: Try to render as much content in plain semantic HTML

    Choice: Option 2:

    Justification: Ideally, we will be using next.js, which should provide us with a good starting framework to create a good working application without expecting javascript in the user's browser. The site does lose a lot of functionality but users can still view content
2. Should we provide an open / well documented REST api to be usable outside our web client?
    a. Option 1: Yes, allows people to build on top of our platform and use our data
    b. Option 2: No

Choice: Option 2

Justification: This cannot be our goal for this project at this stage since having an internal api with no worries of server breakage helps match the velocity of the project. Maybe once every endpoint has been completely finalized this could be looked into.

3. Should we use an external state management library?
    a. Option 1: Yes, use state machines and industry standard best practices to fetch api content, manage user and application state
    b. Option 2: No, this gives us more flexibility on how to model the app

Choice: Option 1

Justification: Using a state machine and data fetching library might have a learning curve but will help as the application scales

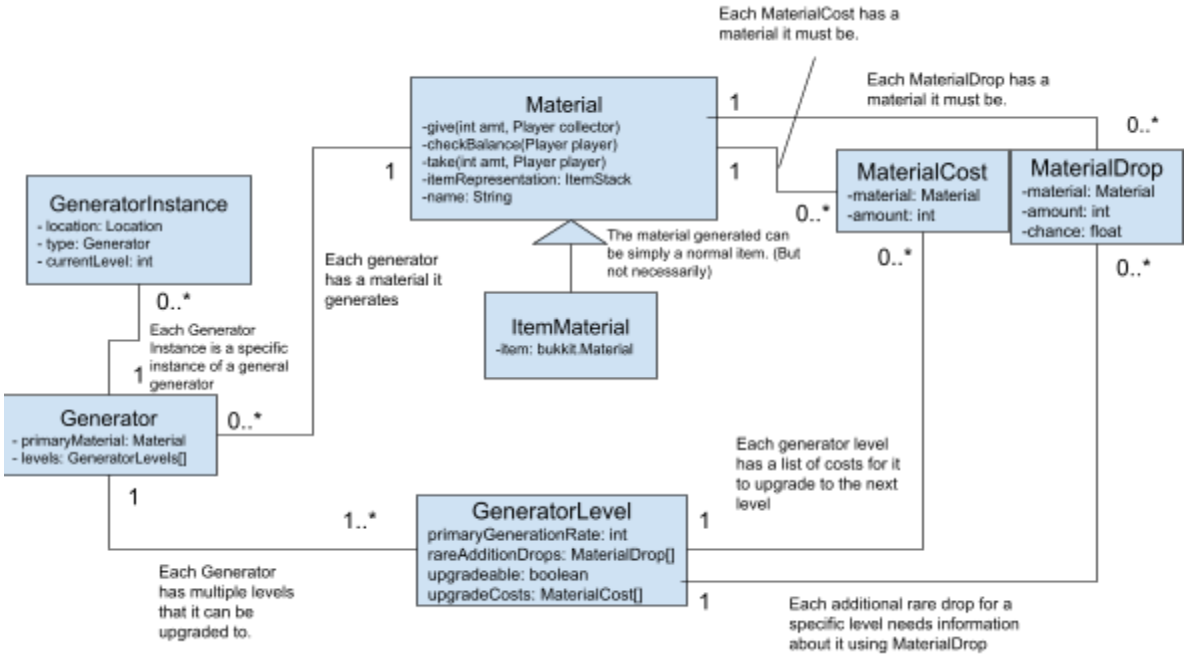
Design Details

Minecraft Server Plugins

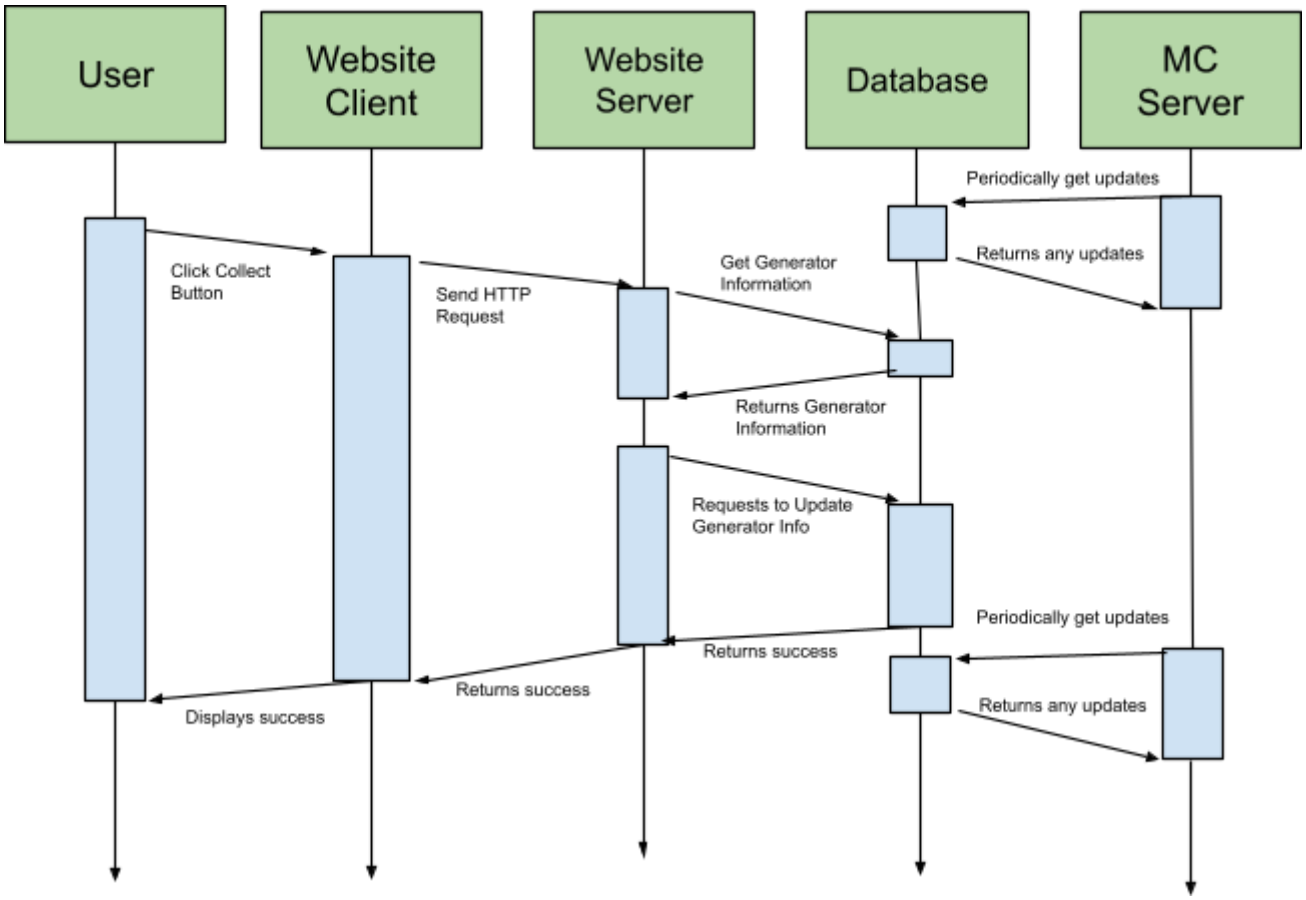
Tycoon Generator System

For the generators, firstly, there will be a large number of generator instances in the world. These will have a specific location with a current level and information about that specific instance, and they will be instances of a specific generator (type). All generator instances of the same generator will have the same characteristics, including the materials it generates, the levels available for it, the generation rates and addition drops at certain levels.

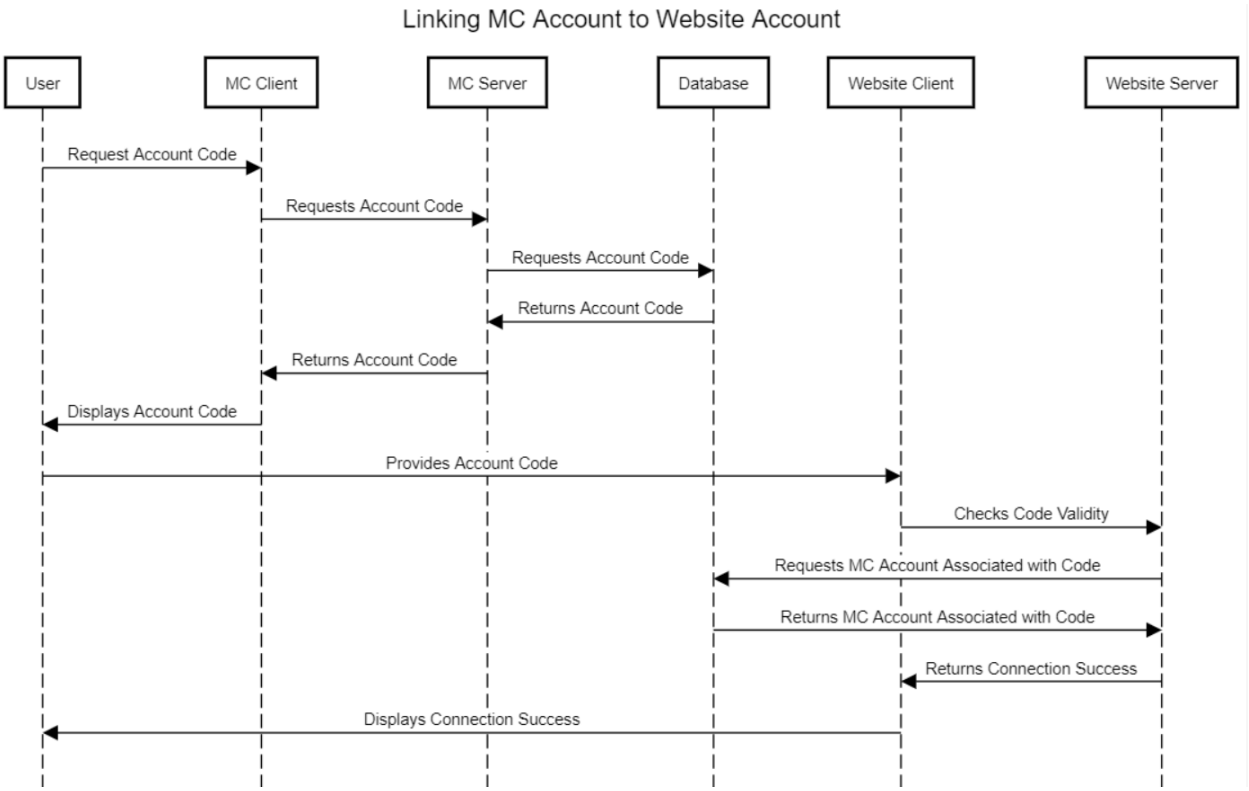
The materials that generators make will either be physical items or abstract materials such as coins. As such, they need to be generalized to 3 potential functions, giving the player an amount, checking if they have an amount (for buying upgrades) and taking a certain amount.



In-game view of generators: Sunflower: Collect, Anvil: Upgrade, Paper: Information



Above figure describes interaction between the website tycoon collection and the Minecraft server.



Above diagram describes the process of a user connecting their Minecraft Account to their Forge Frontier website account.

Gear System

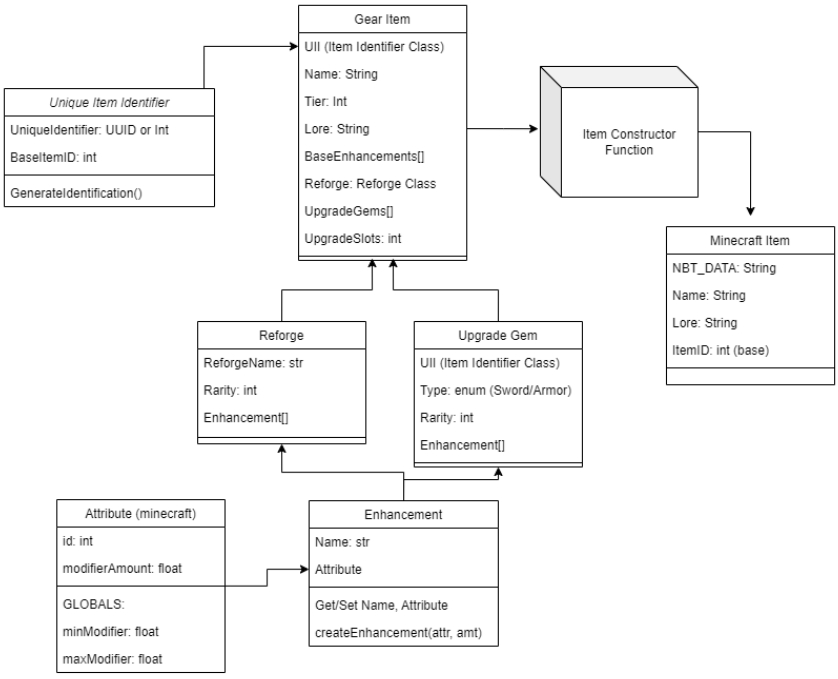
Gear is generated utilizing the Gear Item class, which encompasses all properties a piece of unique gear may have. Gear includes and is not limited to: Armor, Weapons, Tools. Gear has the ability to be reforged and upgraded; each of which provides enhancements. Each piece of gear will also be uniquely identified.

Design Considerations:

1. Prevent duping – Can detect duplicate identifiers and catch duping.

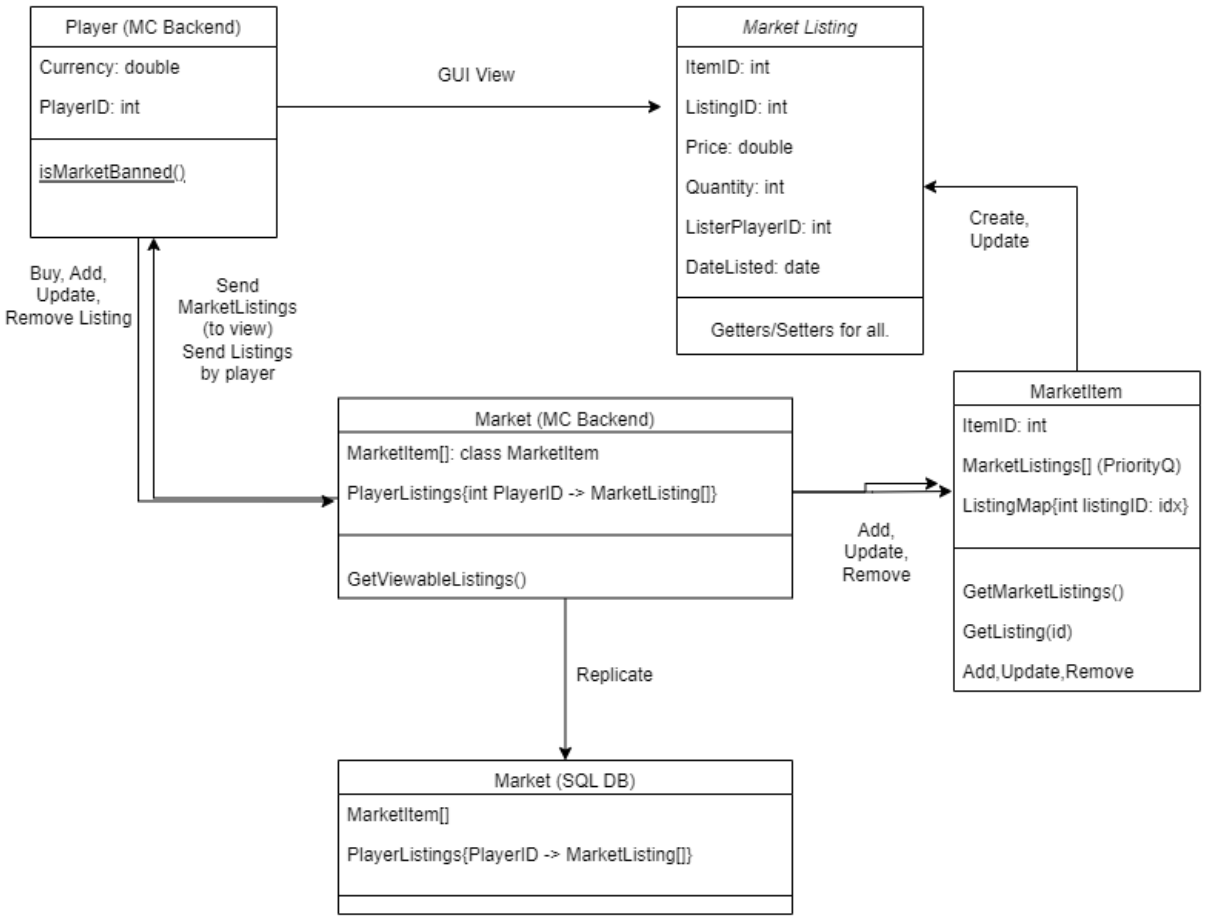
2. Legitimate Gear items are pipelined ONLY through Item Constructor, providing consistency

3. To perform balancing changes, Gear Items will have a tag within the unique identifier which allows for global balancing of said gear; same applies to changes in enhancement.



Global Player Shop Design

The Market is composed of general items (non-unique) for players to buy. Players may list buy or sell orders and be able to complete them. The Market class stores all of this information, and will be replicated to the SQL DB (only replicated one way, MC to DB).



Website Mockup

Market Viewer:

Profile Page:

Idle Clicker Page: