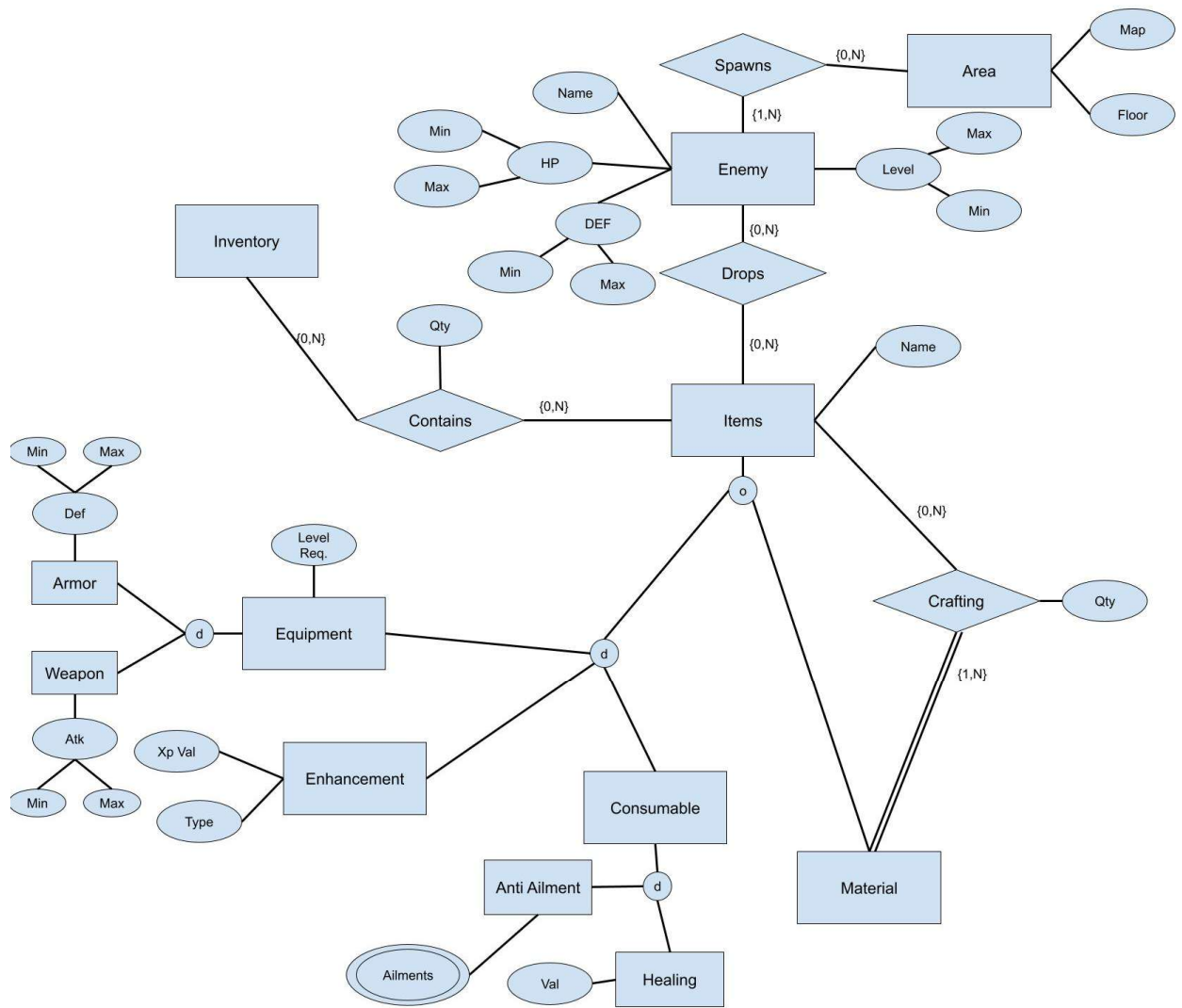The application is a companion system for the video game Sword Art Online Integral factor. The purpose of which is to keep track of not just what items a player has obtained but also their quantity. I want to be able to look up craft able items based on what materials they currently have, and by item attributes such as its type, and level requirements. A database system would be used to keep track of all possible items a player could obtain and craft and to conduct the searches on the items in the database based off a variety of different criteria.
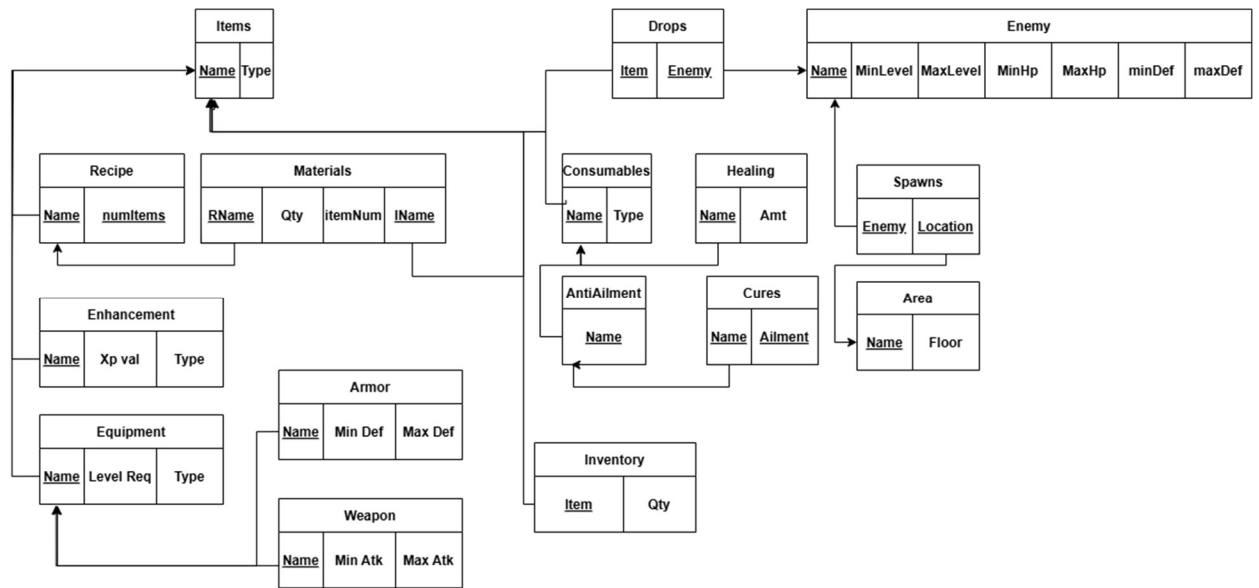
Entity-Relationship Diagram

Spawns {0,N} — **Area** — Map, Floor

Spawns {1,N} — **Enemy**

Enemy attributes:
- Name
- HP (Min, Max)
- DEF (Min, Max)
- Level (Max, Min)

Enemy {0,N} — **Drops** {0,N} — **Items**

Inventory {0,N} — **Contains** — Qty {0,N} — **Items**

Items — Name

Items {0,N} — **Crafting** — Qty

Crafting {1,N} — **Material**

Items (o) — **Equipment** (d) / **Consumable** / **Material**

Equipment — Level Req.

Equipment (d) — **Armor** / **Weapon**

Armor — Def (Min, Max)

Weapon — Atk (Min, Max)

Enhancement — Xp Val, Type

Consumable (d) — **Anti Ailment** / **Healing**

Anti Ailment — Ailments

Healing — Val

Items

Name	Type

Drops

Item	Enemy

Enemy

Name	MinLevel	MaxLevel	MinHp	MaxHp	minDef	maxDef

Recipe

Name	numItems

Materials

RName	Qty	itemNum	IName

Consumables

Name	Type

Healing

Name	Amt

Spawns

Enemy	Location

Enhancement

Name	Xp val	Type

AntiAilment

Name

Cures

Name	Ailment

Area

Name	Floor

Armor

Name	Min Def	Max Def

Equipment

Name	Level Req	Type

Inventory

Item	Qty

Weapon

Name	Min Atk	Max Atk

Table	Attribute	Type	Constraint
Items	Name	Char(80)	Primary Key
Items	Type	Char(10)	
Drops	Item	Char(80)	Foreign Key
Drops	Enemy	Char(80)	Foreign Key
Enemy	Name	Char(80)	Primary Key
Enemy	MinLevel	Int	Not Null
Enemy	MaxLevel	Int	Not Null
Enemy	minHp	Int	Not Null
Enemy	maxHp	Int	Not Null
Enemy	minDef	Int	Not Null
Enemy	maxDef	Int	Not Null
Recipe	Name	Char(80)	Foreign Key
Recipe	NumItems	Int	Not Null
Materials	RName	Char(80)	Foreign Key
Materials	Qty	Int	Not Null
Materials	itemNum	Int	Not Null
Materials	IName	Char(80)	Foreign Key
Enhancement	Name	Char(80)	Foreign Key
Enhancement	Xp Val	Int	Not Null
Enhancement	Type	Char(20)	Equipment/Skill
Equipment	Name	Char(80)	Foreign Key
Equipment	Level Req	Int	
Equipment	Type	Char(10)	Armor/Weapon
Drops	Item	Char(80)	Foreign Key
Drops	Enemy	Char(80)	Foreign Key
Consumables	Name	Char(80)	Foreign Key
Consumables	Type	Char(10)	Ailment/healing
Healing	Name	Char(80)	Foreign key
Healing	Amt	Int	Not Null
AntiAilment	Name	Char(80)	Foreign Key
AntiAilment	Ailments	Char(80)	Foreign Key
Cures	Name	Char(80)	Foreign Key
Cures	Ailment	Char(40)	Primary Key
Spawns	Enemy	Char(80)	Foreign Key
Spawns	Location	Char(80)	Foreign Key
Area	Name	Char(80)	Primary Key
Area	Floor	Int	
Inventory	Item	Char(80)	Foreign Key
Inventory	Qty	Int	Not Null

```
Login()

        Name = Uname

        If (Name.db exists)

                User = Name

                Enable Search and Inventory Page

        Else:

                Display "Please create New user"


Create()

        Name = ask user for desired UserName

        While (Name in users)

                Display "Please Choose a different User Name"

                Name = new name

        Create New User database


Craftable(Item)

        mats = "Select * from Materials Where Rname = item"

        If (mats.size()==0):

                Return false

        Else:

                For item in mats:

                        If "Select Qty from Inventory where name=item.name"<item.qty:

                                Return false

Return true
```

```
Craft(item):

        For items in "Select * from Materials where Rname=item":

                Inventory.items.qty -= items.qty

        Inventory.item += 1

addToInventort(item):

        if (item in Inventory):

                return;

        else;

                Inventory.addItem(item)


Inc(item):

        if (not "Select * from Inventory Where Item=item":

                "Insert into Inventory values('item, 1)"

        else:

            "Update Inventory Set Qty =  ("Select Qty from Inventory Where Item='item'")+1}
where Item='{item}'"


Dec(item):

  "Update Inventory Set Qty = ("Select Qty from Inventory Where item='item'")-1} where
Item='item'")

  if ("Select Qty from Inventory Where item='item'")<1:

     "Delete from Inventory Where item='{item}'"
```