Game Store Project

This project involves creating a simple database backed REST inventory management web service for a Video Game Store using Agile development techniques in a solo setting. You are responsible for designing and documenting the REST API and implementing the controller, service, layer, DAO, Java data objects, and unit tests for the application based on an existing database structure.

Structure

Your solution must have the following structural elements:

- Your solution must be in an IntelliJ project called `FirstNameLastNameU1Capstone` where FirstName and LastName are your first and last names respectively.
- Your project must be built using Spring Boot and Spring MVC. Initialize your project using `start.spring.io`
- Your solution must include a DAO the utilized JdbcTemplates and Prepared Statements
- Your REST API must be documented with Swagger
- Your REST API must accept and return data in JSON format where appropriate
- You must implement ControllerAdvice to handle exceptions and return propery HTTP status codes and data when exception occur. This includes handling all violoations of business rules.

Methodology

- You must manage your work in Pivotal Tracker

- You must create stories and epics
- You must estimate your work using story points
- You must use a Test Driven Development approach (inluding Red/Green/Refactor) for your code
- You must use JUnit for unit and integration tests
- Your design must include a Service Layer
- Your unit test suite should utilize mock objects where appropriate
- You should utilize JSR303 for input validation

Requirements/Features

This system must manage the inventory of video games, game consoles, and t-shirts.

- Your REST API must allow the end user to:
  i. Games:
     a. Perform standard CRUD operations for Games
     b. Search for Games by Studio
     c. Search for Games by ESRB Rating
     d. Search for Games by Title
     e. You must create a separate DAO for Games
  ii. Consoles:
     a. Perform standard CRUD operations for Consoles
     b. Search for Consoles by Manufacturer
     c. You must create a separate DAO for Consoles
  iii. T-Shirts:
     a. Perform standard CRUD operations for T-Shirts
     b. Search for T-Shirts by Color
     c. Search for T-Shirts by Size
     d. You must create a separate DAO for T-Shirts
  iv. Purchasing Items:

a. User should be able to purchase items in inventory by supplying the following information to the endpoint:
   a. Name
   b. Street
   c. City
   d. Zip
   e. Item Type
   f. Item ID
   g. Quantity
b. The endpoint returns invoice data based on the invoice table below.
c. All invoice calculations must be done in the Service Layer.
d. You must create a DAO for both taxes and processing fees.

You must use the following database structure:

```
create schema if not exists game_store;
use game_store;

create table if not exists game (
    game_id int(11) not null auto_increment primary key,
    title varchar(50) not null,
    ersb_rating varchar(50) not null,
    description varchar(255) not null,
    price decimal(5, 2) not null,
    studio varchar(50) not null,
    quantity int(11)
);

create table if not exists console (
    game_id int(11) not null auto_increment primary key,
    model varchar(50) not null,
    manufacturer varchar(50) not null,
    memory_amount varchar(20),
    processor varchar(20),
    price decimal(5, 2) not null,
    quantity int(11) not null
);

create table if not exists t_shirt (
     t_shirt_id int(11) not null auto_increment primary
key,
    size varchar(20) not null,
    color varchar(20) not null,
    description varchar(255) not null,
    price decimal(5,2) not null,
    quantity int(11) not null
);

create table if not exists sales_tax_rate (
     state char(2) not null,
    rate decimal(3,2) not null
);

create unique index ix_state_rate on sales_tax_rate
(state, rate);

create table if not exists processing_fee (
     product_type varchar(20) not null,
    fee decimal (4,2)
);

create unique index ix_product_type_fee on processing_fee
(product_type, fee);

create table if not exists invoice (
    invoice_id int(11) not null auto_increment primary
key,
    name varchar(80) not null,
    street varchar(30) not null,
    city varchar(30) not null,
    state varchar(30) not null,
    zipcode varchar(5) not null,
    item_type varchar(20) not null,
    item_id int(11) not null,
    unit_price decimal(5,2) not null,
    quantity int(11) not null,
    subtotal decimal(5,2) not null,
    tax decimal(5,2) not null,
    processing_fee decimal (5,2) not null,
    total decimal(5,2) not null
);
```

Business Rules

1. Sales tax applies only to the cost of the items.
2. Sales tax does not apply to any processing fees for an invoice.
3. The processing fee is applied only once per order regardless of the number of items in the order unless the number of items on the order is greater than 10 in which case an *additional* processing fee of $15.49 is applied to the order.
4. The order process logic must properly update the quantity on hand for the item in the order.
5. Order quantity must be greater than zero.
6. Order quantity must be less than or equal to the number of items on hand in inventory.
7. Order must contain a valid state code.

8. The REST API must properly handle and report all violations of business rules.

Data

Tax Rates

Load the following tax rates into your database:

- Alabama - AL: .05
- Alaska - AK: .06
- Arizona - AZ: .04
- Arkansas - AR: .06
- California - CA: .06
- Colorado - CO: .04
- Connecticut - CT: .03
- Delaware - DE: .05
- Florida - FL: .06
- Georgia - GA: .07
- Hawaii - HI: .05
- Idaho - ID: .03
- Illinois - IL: .05
- Indiana - IN: .05
- Iowa - IA: .04
- Kansas - KS: .06
- Kentucky - KY: .04
- Louisiana - LA: .05
- Maine - ME: .03
- Maryland - MD: .07
- Massachusetts - MA: .05
- Michigan - MI: .06
- Minnesota - MN: .06
- Mississippi - MS: .05

- Missouri - MO: .05
- Montana - MT: .03
- Nebraska - NE: .04
- Nevada - NV: .04
- New Hampshire - NH: .06
- New Jersey - NJ: .05
- New Mexico - NM: .05
- New York - NY: .06
- North Carolina - NC: .05
- North Dakota - ND: .05
- Ohio - OH: .04
- Oklahoma - OK: .04
- Oregon - OR: .07
- Pennsylvania - PA: .06
- Rhode Island - RI: .06
- South Carolina - SC: .06
- South Dakota - SD: .06
- Tennessee - TN: .05
- Texas - TX: .03
- Utah - UT: .04
- Vermont - VT: .07
- Virginia - VA: .06
- Washington - WA: .05
- West Virginia - WV: .05
- Wisconsin - WI: .03
- Wyoming - WY: .04

Processing Fees

Load the following processing fees into your database:

- Consoles: 14.99
- T-Shirts: 1.98
- Games: 1.49