Activity 3

Daniel R. Cender

Grand Canyon University: CST-256-O500

Jan 14, 2021

**Repository Info**

**GitHub Repository**

The project is hosted at: <https://github.com/DanielCender/CST-256>. This assignment’s code is contained under the folder ‘activity3’. No vendor information was pushed with the project code, so all Composer dependencies will need to be installed prior to running ‘php artisan serve’.

**Data Validation**

This section went over defining some basic form validation and returning error codes to the blade template view.

**Screenshots**

1. Login Form with Username validation errors

Graphical user interface, text, application, email

Description automatically generated

1. Login Form with Password validation errors

Graphical user interface, text, application, email

Description automatically generated

**ACID Database Transactions**

This section introduces using transaction-based MySQL queries to chain together data operations into a larger operation.

**Screenshots**

First photo shows the result of the /customer/create API route controller actions.

Graphical user interface, text

Description automatically generated

Second photo shows the result of the /order/create API route controller actions.Graphical user interface, text

Description automatically generated

Third screenshot shows the result of the transaction-powered /order/create API route handler. The two tables are queried in a join to show the new data from both tables.

Graphical user interface, text

Description automatically generated

**Component Diagram**

This basic diagram shows how the different pieces of this service all work. The OrderService interfaces with both the Customer DAO and Order DAO to create a new Customer record before creating an associated Order.

Diagram

Description automatically generated

**ACID Explanation**

The ordering of database operations in this improved order creation model allows the application to support the tenants of ACID. By using MySQL transactions to package the Customer and Order creation process, it’s ensured that our operation will adhere to the “ID” of ACID (Isolation and Durability). Those two principles are upheld by the database engine itself. In opening a transaction, we isolate the data operations until a commit or rollback action. Once the transaction returns, we can trust the database engine has returned the true state of the data tables. By checking for errors in the data inserts, we’ll be rolling back any “partial” data changes that may result in incomplete or orphaned records. The data will therefore be consistent, since we’ll be able to ensure the stability of the cascade delete behavior placed on the Orders table.

References