Test Plan

Introduction

Unit test is used to test this project. This project consists of multiple modules. For each method in each module, a unit test is designed to make sure it works as expected. The implementation is JUnit5 in Java.

Scope

The test contains the following major categories:

Database Basic Tests

- Table creation and drop for “InventoryRecord", "OrderRecord", "OrderTable","Product", "Customer".
- Trigger creation and drop for createOrderRecordTrigger and deleteOrderRecordTrigger. createOrderRecordTrigger and deleteOrderRecordTrigger guarantee the correct addition and deduction of quantity in stock.
- Function creation and drop for isSku. isSku is used to validate the ProjdectSKU.

Services Tests

CustomerService:

Method	Condition
testInsert	A valid Customer including name, address, city, state, country and postalCode to be inserted. All of the fields are non-null.
testUpdate	A Customer in database and another Customer with the same customer id but different in any of name, address, city, state, country and postalCode to update the one in database. All of the fields are non-null.
testDeleteById	A valid Customer including name, address, city, state, country and postalCode to be inserted. All of the fields are non-null.
testGetById	A valid Customer including name, address, city, state, country and postalCode to be inserted. All of the fields are non-null.

InventoryRecordService:

Method	Condition
testInsert	A valid InventoryRecord including quantity in stock, unit price and sku. All of the fields are non-null. Quantity and price are positive and price has 2 digits after the decimal place.
testGetAll	Severa validl InventoryRecord in database.
testAddInvalidInventoryRecord	A valid InventoryRecord but without corresponding Product. A InventoryRecord with non-positive quantity. A InventoryRecord with non-positive price.
testUpdate	A valid InventoryRecord in the database. Another InventoryRecord with the same SKU but different other fields.
testGetById	A valid InventoryRecord including quantity in stock, unit price and sku. All of the fields are non-null. Quantity and price are positive and price has 2 digits after the decimal place.

OrderRecordServiceTest:

Method	Condition
testInsert	A valid OrderRecord including order id, number of units unit price and sku and its corresponding valid Order and Product. All of the fields are non-null. The number of units is less the one in corresponding InventoryRecord.
testInsert_exceedInStock	A valid OrderRecord including order id, number of units unit price and sku and its corresponding valid Order and Product. But the number of units exceeds the one in the corresponding InventoryRecord.
testGetById	A valid OrderRecord including order id, number of units unit price and sku and its corresponding valid Order and Product. All of the fields are non-null.
testGetAll	Several valid OrderRecords including order id, number of units unit price and sku and its corresponding valid Order and Product. All of the fields are non-null.

OrderServiceTest:

Method	Condition
testInsert	A valid Order including customer id, order id, order date, ship date and its corresponding valid Product and OrderRecord. All of the fields are non-null except for ship date.
insertTest_noOrderRecord	A valid Order including customer id, order id, order date, ship date and its corresponding valid Product. All of the fields are non-null except for ship date and OrderRecord.
testShipOrder	A valid Order in database including customer id, order id, order date, ship date and its corresponding valid Product and OrderRecord. All of the fields are non-null except for ship date. A valid ship date and the order id of the Order.
testGetAllOrders	Several valid Orders including customer id, order id, order date, ship date and its corresponding valid Product and OrderRecord. All of the fields are non-null except for ship date.
testGetById	A valid Order including customer id, order id, order date, ship date and its corresponding valid Product and OrderRecord. All of the fields are non-null except for ship date.

ProductServiceTest:

Method	Condition
testInsert	A valid Product including name, description and sku. All of the fields are non-null. SKU is a 12-character value of the form AA-NNNNNN-CC where A is an upper-case letter, N is a digit from 0-9, and C is either a digit or an uppercase letter.
testGetAllProducts	Several valid Products in database.
testInvalidSku	A valid Product including name, description and sku. All of the fields are non-null. SKU is *NOT* a 12-character value of the form AA-NNNNNN-CC where A is an upper-case letter, N is a digit from 0-9, and C is either a digit or an uppercase letter.