Hello.

I have included this notes in order to explain how i built my main test class.

it can help you understand how i wanted you to check my project, but in any case

feel free to check it as you please.

i have included in the folder the mysql file that will create the mysql database for the inspection.

**!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

**Do not forget to change in the ConnectionPool class the parameters root (the name of your server) and the password (your server's password).**

**!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

i have divided my main test class to different tests, numbered 1-10,

which runs on all of the different functions of my project.

hope you enjoy ☺

the first line in my main test is a login by the admin.

the admin user is relevant for test 1-4.

**test 1:**

i have created a few companies and the admin creates them in the database.

a message will be shown in the console that the relevant company was created.

after the first run of this test you can examine the database and see that

those companies were infect created.

you can now run the same test again and see that the admin cannot create these companies

again.

- messages will be printed in the console for every company 2

the message from the company exception handler and the message from

the exception itself (at this point i can only print it to the console, but in

the future it will be written to a log file).

**test 2:**

i have created a few customers and the admin creates them in the database.

a message will be shown in the console that the relevant customer was created.

after the first run of this test you can examine the database and see that those

customers were infect created.

you can now run the same test again and see that the admin cannot create these customers

again.

2 messages will be printed in the console for every customer -

the message from the customer exception handler and the message from

the exception itself (at this point i can only print it to the console, but in

the future it will be written to a log file).

**test 3:**

the admin will try to update a company in the database.

he first needs to get a list of all the companies in the database in order to know

the id of the desired company, and the create a new instance of that company but

with the updated values and update that instance in the database.

afterwards, he will want to see that the update was successful.

in the console you will see a message that confirms that the update was successful.

**test 4:**

the admin will try to update a customer in the database.

he first needs to get a list of all the customers in the database in order to know

the id of the desired customer, and the create a new instance of that customer but

with the updated values and update that instance in the database.

afterwards, he will want to see that the update was successful.

in the console you will see a message that confirms that the update was successful.

**test 5:**

now we have a company user that is logging in.

the first attempt is not correct, since we updated this companies password.

the second attempt is successful.

now the company is trying to create a few coupons.

a message will be shown in the console that the relevant coupon was created.

you can now run the same test again and see that the company cannot create the same

coupon again.

* messages will be printed in the console for every coupon 2

the message from the coupon exception handler and the message from

the exception itself (at this point i can only print it to the console, but in

the future it will be written to a log file).

another message will be shown - because i written a coupon that is out of date,

the daily task will remove it from the database and will show a message accordingly.

next the company want to see his coupons -

all of the coupons of the type food

all of the coupons up to a price of 40

all of the coupons up to a certain end date.

**test 6 :**

now the company want to update a coupon.

first the user will select all of his coupons and get the relevant id.

now he will get the coupon instance to see the values of this coupon.

next he will create an instance of a coupon with different end date and price,

and will update it in the database.

a message will print in the console that the update was successful,

and now the user will get the coupon instance from the database to see the changes.

**test 7 :**

another company user will log in and will create 2 more coupons(not actually a test, just to place more coupons in the system).

**test 8:**

a customer tries to login with a wrong password.

2 messages will be printed in the console -

the message from the customer exception handler and the message from

the exception itself (at this point i can only print it to the console, but in

the future it will be written to a log file).

next the customer will log in with the correct password and will purchase a coupon.

a message will be printed in the console saying the purchase took place.

next, the customer will try to see all of his purchased coupons,

all of his purchased coupons by a certain type and all of his purchased coupons up to a certain price.

**test 9:**

now a company will log in and get a coupon instance from the database and will remove it from the database.

a message will print in the console that the coupon has been removed.

**test 10:**

first step is to login again with a customer and buy 2 coupons.

next, the admin logs in and removes a company and then removes a customer.

the correct message will print to the console.

you can now look in the database and see that all relevant entries

has been removed.

thank you for checking my project.

i hope everything went well.

goodbye ☺