Method Design

Class: CUSTOMERSERVICE

Public Customer Login(Customer customer)

Use to customer log in e-shopping website system. System validate the format of customer username and password. If the validation is correct, then system validate the username and password match each other or not. If match each other, the customer can log in.

Special Case:

* User name format is wrong, prompt the error message is “The user name must be email”
* User password format is wrong, prompt the error message is “The password length should be 5-20”
* User name and password not match, prompt the error message is “ The user name and password is not match”

Parameter:

A customer contains user enter user name and password.

Returns:

The customer which customer login.

Public List<Product> SearchProductByName(String productName)

Get the products that contains same name the customer required.

Special Case:

* If no product has same product name, return an empty list.

Parameter:

The product name can have number, character, and space.

Returns:

A list of all product name same as customer required.

Public void AddToShoppingCart(Product product)

Put the selected product to current customer shopping cart. The shopping cart shall add this new product, and sum of charge shall plus new product’s price.

Special Case:

* If the product already be contained in the shopping cart, the product will not add twice.

Parameter:

A product.

Returns:

None.

Public Customer CheckOut(Customer customer)

Calculate the total of all product price inside the current shopping cart and clear customer shopping cart for next shop.

Special Case:

* If the shopping cart not contains any product, the shopping cart will not check out.

Parameter:

The current customer.

Returns:

The current check out customer.

Public Customer SaveShoppingCart(Customer customer)

Save customer current shopping cart and all products inside, wait the customer check out next time.

Parameter :

The current customer;

Returns:

The current save shopping cart customer.

Public Customer EditProfile(Customer customer)

The customer shall edit the detail self-information for prepare the shopping.

Parameter:

The current customer.

Returns:

The current edit profile customer.

Public Customer Register(Customer customer)

The user register an account for to be a customer of e-shopping website system.

Parameter:

The user not register to e-shopping website.

Returns:

The current register customer.

Public Customer getCurrentCustomer()

Get the customer that login the e-shopping website currently.

Returns:

The current customer.

Public Costomer setCurrentCustomer(Customer customer)

Setting the current e-shopping website system customer.

Parameter:

Customer should be the current login successfully customer.

Returns:

The current Customer.

Public ShoppingCart getSaveShoppingCart(Customer customer)

Get a shopping cart that the customer saved and not finish the check out.

Parameter:

The current customer;

Returns:

The customer lastly saved shopping cart.

Class: ADMINISTRATORSERVICE

Public Product EditProductInfo(Product product)

The admin update the information of the products of website system.

Return a new product that admin have changed.

Public List<ShoppingHistory> ScanShoppingHistory()

The admin can Scan all of customers shopping History.

Returns:

A List<ShoppingHistory> that contains all of customer shopping history.

Public Administrator Login(Administrator admin)

The log into e-shopping website system for customer specially. System validate the format of customer username and password. If the validation is correct, then system validate the username and password match each other or not. If match each other, the customer can log in.

Special Case:

* User name format is wrong, prompt the error message is “The user name must be email”
* User password format is wrong, prompt the error message is “The password length should be 5-20”
* User name and password not match, prompt the error message is “ The user name and password is not match”

Parameter:

An administrator contains user enter user name and password.

Returns:

Current successfully login administrator.

Class CustomerDAO

Public Customer getCustomerById(String customerID)

Get a customer by the customer id.

Parameter:

CustomerID only can contains number.

Returns:

A customer that have same id.

Public Customer addCustomer(Customer customer)

Add a new customer to system from the parameter.

Parameter:

The customer who is new register, need add to system database;

Returns:

The new added customer.

Public Customer updateCustomer(Customer customer)

Update the information inside of current Customer.

Parameter:

The customer that contains new customer information which will recover the old customer.

Returns:

The new changed Customer.

Public List<Customer> getCustomers()

Get the all customers of e-shooping website system.

Special Case:

* If there are no any customer in system, it will be return an empty list.

Returns:

All of customers.

Class :ShoppingCartDAO

Public shoppingCart getShoppingCart(Customer customer)

Get a customer’s shopping cart with the parameter.

Parameter:

The current customer;

Returns:

The special customer’s shopping cart.

Public ShoppingCart addProduct(Product product)

Put the product that customer chose to the customer’s shopping cart.

Parameter:

The product user chose.

Returns:

The current customer’s shoppingCart.

Public ShoppingCart DeleteProduct(Product product)

Delete the product that inside the current customer shopping cart.

Parameter:

Product should be the product that has be inside the current customer shopping cart.

Returns:

The shopping cart that have delete changed.

Public List<Product> getProducts(ShoppingCart shoppingCart)

From the parameter shopping cart, get the all product that inside the shopping cart.

Parameter:

The current customer’s currently shopping cart.

Returns:

A List<Product> that contains all product in shopping cart.

Public ShoppingCart clearShoppingCart(ShoppingCart shoppingCart)

Delete all products in the shopping cart.

Parameter:

The current customer’s current shopping cart;

Returns:

An empty shopping cart.

Class: PRODUCTDAO

Public Product getProductByName(String productName)

Get a product that have same name with parameter.

Parameter:

The product name can contains the number, character, and space;

Returns:

A product that have same name.

Public Product getProductByID(String productID)

Get a product that have same id with the parameter.

Parameter:

The product id only can contains number;

Returns:

A product that caontains same id.

Public Product addProduct(Product product)

Add a new product to the e-shopping website system.

Parameter:

The product is not inside the system.

Returns:

The new product.

Public boolean DeleteProduct(Product product)

Delete a product in the e-shopping website system.

Parameter:

The product that be contained in the current customer’s shopping cart currently;

Returns:

The true, the product delete successful.

Public product updateProduct(Product product)

Change a product of e-shopping websie system.

Parameter:

The product that inside the system.

Returns:

The changed new product.

Class: ShoppingHistoryDAO

Public ShoppingHistory getShoppingHistoryById(String customerID)

Get the shopping history belong to special customer that customer id same as the parameter.

Parameter:

The customer is only can contains the number;

Returns:

The special customer’s shopping history.

Public List<ShoppingHistory> getShoppingHistories()

Get all of customers shopping histories in the e-shopping website.

Parameter:

None;

Returns:

A List<ShoppingHistory> that contains all shopping histories.

Public ShoppingHistory AddShoppingHistory(ShoppingCart shoppingCart)

Make a shopping cart that have check out to be a shopping history with the customer name and customer id;

Parameter: The Shopping cart that the customer has check out

.

Returns:

The new shopping history.

ClassName: Register

Public Boolean validateUsername(String username)

Validate the format of username before user register a new account;

Parameter:

The user name that user to be login website to distinguish identity of user.

Username format should be email format.

Return:

If the username is standard email format, it’s shall return true;

If the username is not satisfy the email format, it’s shall return false;

Public Boolean validatePassword(String password)

Validate the format of user password before the user register a new account;

Parameter:

The password that check this user is real user of website.

Password format is 5 to 10 any characters.

Return:

If the password is satisfy the format, it’s shall return true;

If the password is not satisfy the format, it’s shall return false;

Public Boolean validateRepassword(String rePassword)

Validate the repassword is password format or not, and check repassword is same as password or not.

Parameter:

To check the password again.

Repassword shall be same as the password.

Return:

If the repassword is same as password, it’s return true;

If the repassword is not same as password, it’s return false;

Class: Login

Public Boolean validateUsername(String username)

Validate the format of username before check the username match with user password;

Parameter:

The password that check this user is real user of website.

Username format should be email format.

Return:

If the password is satisfy the format, it’s shall return true;

If the password is not satisfy the format, it’s shall return false;

Public Boolean validatePassword(String password)

Validate the format of user password before check the username match with user password;

Parameter:

The user name that user to be login website to distinguish identity of user.

Password format is 5 to 10 any characters.

Return:

If the username is standard email format, it’s shall return true;

If the username is not satisfy the email format, it’s shall return false;