7640 group project

Online Retail Database Application

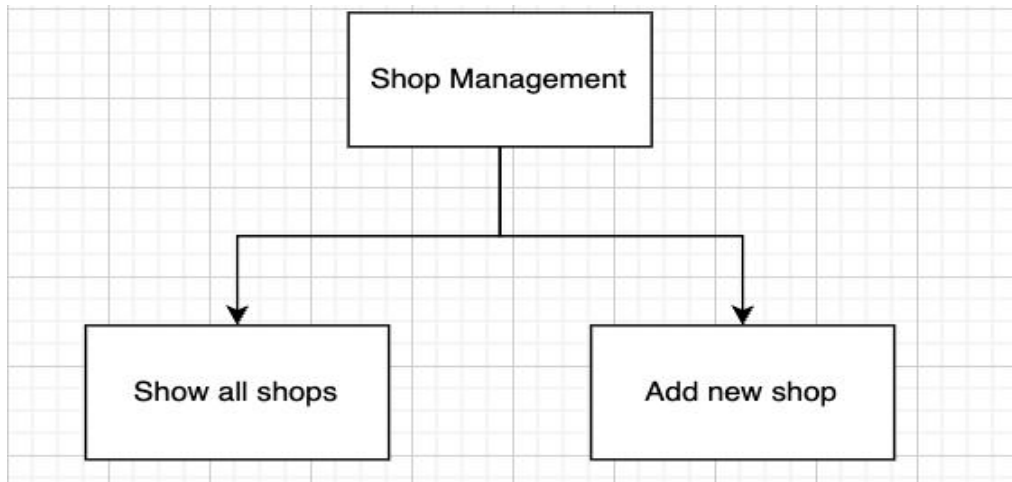## Group (32) Members:

Ye Hanyi 22414495

Wu Chen 22443606

Qiu Chen 22427481
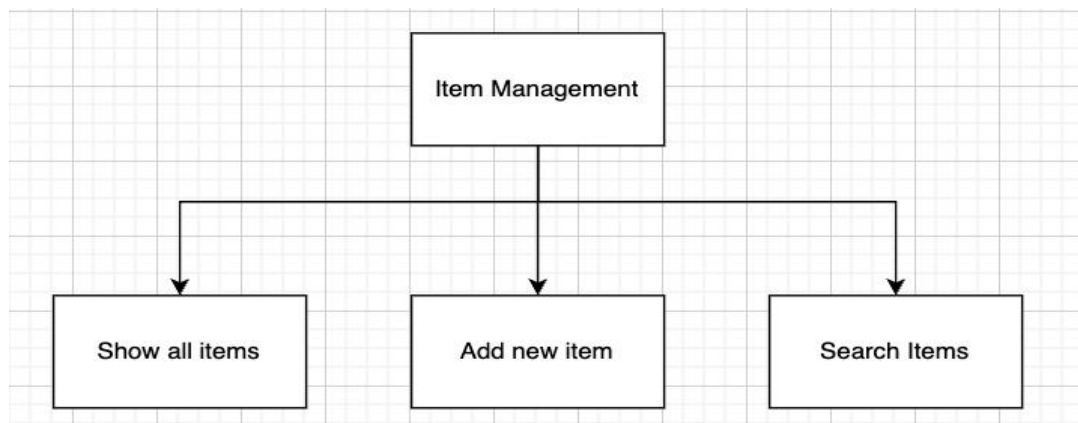
Zhang Chi 22407987

1. System Functionality Design

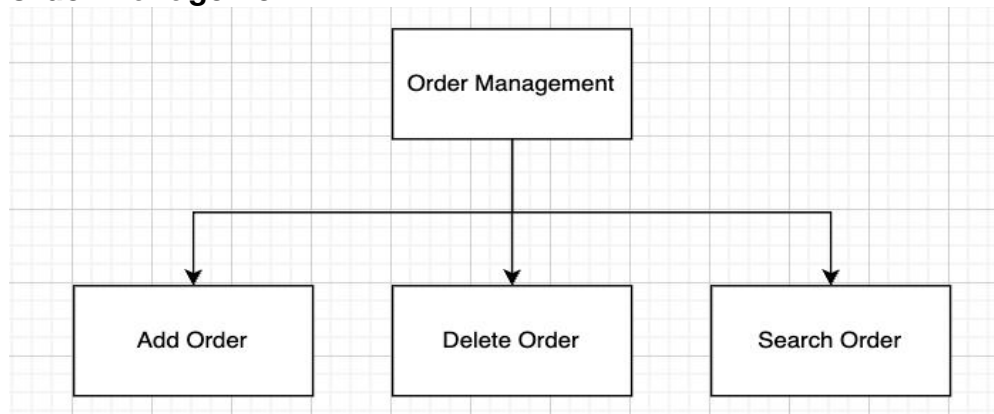## (1) Shop Management:

The functions of the shop management module include shop information query and insert new shop information.
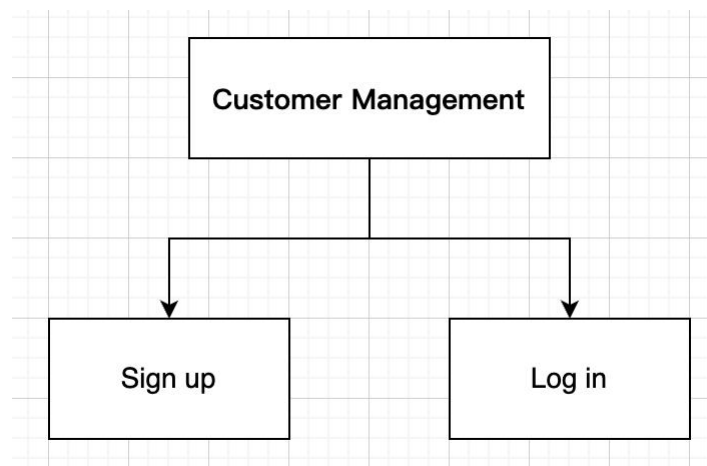
(2) Item Management:

The functions of the item management module include item information query, search and insert new item information.

(3) Order Management

The functions of the order management module include order information query, insert and delete order information.

(4) Customer Management

The functions of the customer management module include sign up and log in.


2. System Detail Design

(1) Database Design

This Online Retail Database Application is developed by using Vue.js, JS, TS and Java. And the data are stored in Mysql. Main data include Shop Data, Item Data, Customer Data, Order Data. The relationships among these data are shown below:

1) a vendor can have many products, a product can be stored in many vendors. (m-m)
2) a product can have many productnotes, a productnote can be used to describe many product. (m-m)
3) an order can have many orderitems, a orderitem can be bought by many order. (m-m)
4) an orderitem can have many products, a product can exist in many orderitems. (m-m)
5) a customer can create many orders, an order can be created by only a customer. (1-m)

(2) ER Diagram
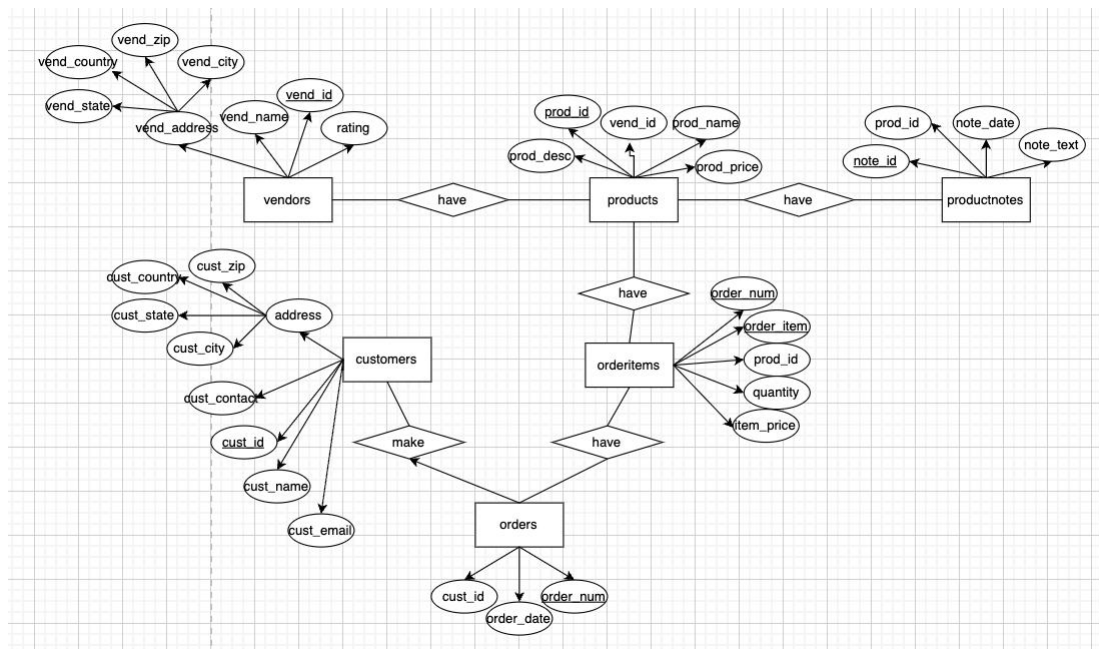
(3) Relational Tables

Column	Type	Nullable	Indexes
◇ cust_id	int	NO	PRIMARY
◇ cust_name	char(50)	NO	
◇ cust_address	char(50)	YES	
◇ cust_city	char(50)	YES	
◇ cust_state	char(5)	YES	
◇ cust_zip	char(10)	YES	
◇ cust_country	char(50)	YES	
◇ cust_contact	char(50)	YES	
◇ cust_email	char(255)	YES	

1 tb_customers

Column	Type	Nullable	Indexes
◇ order_num	int	NO	PRIMARY
◇ order_item	int	NO	PRIMARY
◇ prod_id	char(10)	NO	fk_orderitems_products
◇ quantity	int	NO	
◇ item_price	decimal(8,2)	NO	

2 tb_orderitems

Column	Type	Nullable	Indexes
order_num	int	NO	PRIMARY
order_date	datetime	NO	
cust_id	int	NO	fk_orders_customers

3 tb_orders

Column	Type	Nullable	Indexes
note_id	int	NO	PRIMARY
prod_id	char(10)	NO	
note_date	datetime	NO	
note_text	text	YES	note_text

4 tb_productnotes

Column	Type	Nullable	Indexes
prod_id	char(10)	NO	PRIMARY
vend_id	int	NO	fk_products_vendors
prod_name	char(255)	NO	
prod_price	decimal(8,2)	NO	
prod_desc	text	YES	

5 tb_products

Column	Type	Nullable	Indexes
vend_id	int	NO	PRIMARY
vend_name	char(50)	NO	
vend_address	char(50)	YES	
vend_city	char(50)	YES	
vend_state	char(5)	YES	
vend_zip	char(10)	YES	
vend_country	char(50)	YES	

6 tb_vendors

(4) Function Realization
1) Shop Management:

HomePage			
Item service			
Shop service			

Shops List

shopName	location	Operations
Ass	CN	ℹ Details
ZAS	AM	ℹ Details
FAA	UK	ℹ Details
WSS	TW	ℹ Details
PAA	HK	ℹ Details
WFZ	CN	ℹ Details
Ass	CN	ℹ Details
ZAS	AM	ℹ Details
FAA	UK	ℹ Details
WSS	TW	ℹ Details
PAA	HK	ℹ Details
WFZ	CN	ℹ Details

return to home page

customer-13

Items info in this shop ✕

itemName	price	shopId	keywords
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work

HomePage		
Item service		
Shop service		

customer-13

add shop

shopName [ ]

location [ ]

You don't have any shop

add shop

shopName AMAZON

location US

Shop	Add Items
Shop name: AMAZON	
location: US	
Item List	>

```java
@RestController
@RequestMapping("/shop")
@Api(value = "Shop API")
public class ShopController {

    @Autowired
    ShopService shopService;

    @GetMapping("/test")
    public R test() {
        return R.success();
    }

    @ApiOperation("Show all the shops")
    @GetMapping("/showShops")
    public R showShops() {
        return R.success(shopService.showShops());
    }

    @ApiOperation("Add a shop")
    @PostMapping("/addShop")
    public R addShop(@RequestParam(value = "shopName") String shopName,
                     @RequestParam(value = "location") String location) {
        if (shopService.addShop(shopName, location)) {
            return R.success();
        } else {
            return R.error(REnum.ADD_SHOP_FAILED);
        }
    }

}
```

2) Item Management:

Shop	Add Items

Shop name: AMAZON

location: US

Item List ⌄

itemName	price	shopId	keywords
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work

HomePage

Item service

Shop service ⌄

Shop

Shop name: AMAZON

location: US

Item List

itemName	price	shopId	keywords
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work
Glass	30	6	home work

itemName(*)

price(*)

keyword1

keyword2

keyword3

continue

HomePage

Item service

Shop service ⌄

customer-13

Please input

Select ⌃

ItemKeyword

ShopId

itemName	price	shopId	keywords	Operations
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart
Glass	30	6	home work	Add to shoppingCart

return to home page

```
@RestController
@RequestMapping("/item")
public class ItemController {

    @Autowired
    ItemService itemService;

    @ApiOperation("Add an item to a shop")
    @PostMapping("/addItem")
    public R addItem(@RequestParam Integer shopId, @RequestParam String itemName,
                     @RequestParam BigDecimal price, @RequestParam(required = false) String keyword1,
                     @RequestParam(required = false) String keyword2, @RequestParam(required = false) String keyword3) {
        if (itemService.addItem(shopId, itemName, price, keyword1, keyword2, keyword3)) {
            return R.success();
        } else {
            return R.error(REnum.ADD_ITEM_FAILED);
        }
    }

    @ApiOperation("Show items")
    @GetMapping("/showItems/{shopId}")
    public R showItems(@PathVariable Integer shopId) {
        return R.success(itemService.showItems(shopId));
    }

    @ApiOperation("Search items")
    @GetMapping("/searchItems")
    public R searchItems(@RequestParam String input) {
        return R.success(itemService.searchItems(input));
    }

}
```

3) Order Management

customer-13

	itemName	price	shopId	keywords
☐	glass	30	1	nice well
☑	toy	40	1	bad
☐	T-shirt	60	2	quality common
☑	shoe	100	3	expensive
☐	butter	20	3	tasty favour great
☑	glass	30	1	nice well
☑	toy	40	1	bad
☐	T-shirt	60	2	quality common
☐	shoe	100	3	expensive
☐	butter	20	3	tasty favour great
☑	glass	30	1	nice well
☐	toy	40	1	bad

[Cancel order(s)] [Clear selection]

return to shoppingCart

```java
@RestController
@RequestMapping("/order")
public class OrderController {

    @Autowired
    OrderService orderService;

    @ApiOperation("Make a new order, which may consist of several items")
    @PostMapping("/newOrder")
    public R newOrder(HttpSession session, @RequestBody OrderVo orderVo) {
        orderService.newOrder((Integer) session.getAttribute("customerId"), orderVo.getItems());
        return R.success(orderVo);
    }

    @ApiOperation("Remove order, which may not contains all items in an order")
    @PostMapping("/cancelOrder")
    public R cancelOrder(@RequestBody CancelOrderVo cancelOrderVo) {
        orderService.cancelOrderItems(cancelOrderVo.getOrderId(), cancelOrderVo.getItems());
        return R.success();
    }

}
```

```java
@RestController
@RequestMapping("/database/orderItem")
public class OrderItemController {

}
```

4) Customer Management

Sign in

Fill in the customerID to sign in

CustomerId

CustomerId

Continue

New to here?

Create new one

COMP7640 G32

YE Hanyi 22414495

ZHANG Chi 22407987

WU Chen 22443606

QIU Chen 22427481

Sign up

Fill in the Infos to sign up

Telephone

telephone

Address

address

continue   reset

Already have sign up?

Sign in

COMP7640 G32

YE Hanyi 22414495

ZHANG Chi 22407987

WU Chen 22443606

QIU Chen 22427481

```java
@RestController
@RequestMapping("/")
public class CustomerController {

    @Autowired
    CustomerService customerService;

    @ApiOperation("Sign up")
    @PostMapping("/signUp")
    public R signUp(HttpSession session, @RequestParam String telephone, @RequestParam String address) {
        int cId = customerService.signUp(telephone, address);
        session.setAttribute("customerId", cId);
        return R.success(cId);
    }

    @ApiOperation("Sign in")
    @PostMapping("/signIn")
    public R signIn(HttpSession session, @RequestParam Integer customerId) {
        if (session.getAttribute("customerId") != null && session.getAttribute("customerId").equals(customerId)) {
            return R.error(REnum.LOGIN_BEFORE);
        } else if (customerService.isValidCustomer(customerId)) {
            session.setAttribute("customerId", customerId);
            return R.success();
        } else {
            return R.error(REnum.NO_CUSTOMER_ERROR);
        }
    }

    @ApiOperation("Log out")
    @PostMapping("/logOut")
    public R logOUt(HttpSession session) {
        Integer cId = (Integer) session.getAttribute("customerId");
        if (cId != null) {
            session.removeAttribute("customerId");
            return R.success();
        }
        return R.error(REnum.NOT_LOGIN_ERROR);
    }

}
```