

Submission 3

Keran Wang (keranw 686976)

Xue Jiang (jiangx2 665401)

1. Use cases

1.1 Use case for feature A

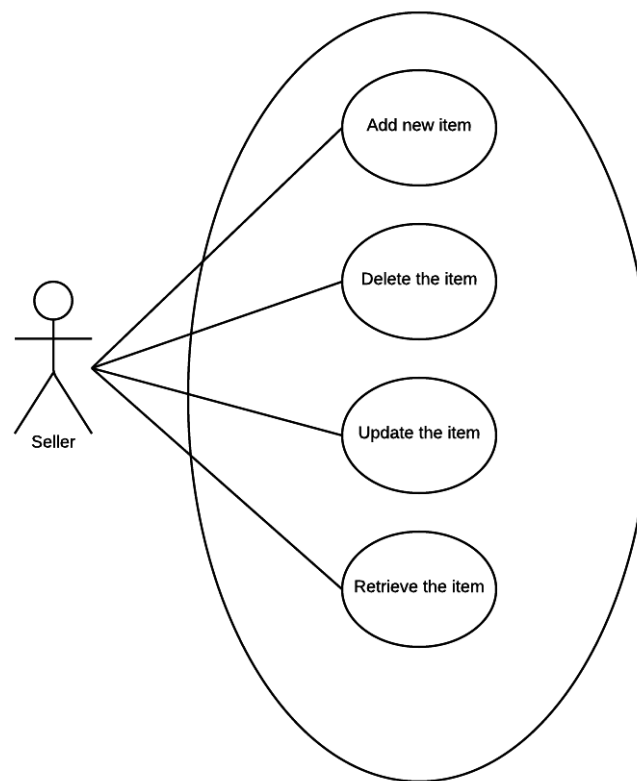


Figure 1.1 Use case diagram for feature A

1.1.1 Add new items

ID	UCA01
Description	The seller can add new items into the system to sale.
Precondition	The user logged in with a seller account
Basic flow	1. The seller clicks the "add a new item" button on the home page. 2. The system provides a form page for the user. 3. The seller fills the form with the information about the new item and clicks the save button to submit. 4. The system receives the form and saves the new record into the database. 5. The system redirects to the home page.
Alternative flow	3a. The user clicks the cancel button to give up add new item. 3a1. The system redirects to the home page. 4a. The information filled in the form is incorrect and the system alert a message to the user. 4a1. The system keeps in the edit page waiting for the user

	to correct the information.
Post condition	The record about the new item is saved in the database.

1.1.2 Retrieve items

ID	UCA02
Description	The user checks the information about the items.
Precondition	The user logged in with a seller account and has added at least one item into the system.
Basic flow	<ol style="list-style-type: none"> 1. The user chooses to search all the items in the database. 2. The system returns a list of the items which are added by the user in the database. 3. The user selects one item in the list by clicking the "Details" button. 4. The system directs to a new page to show the detail information about the item.
Alternative flow	None
Post condition	None

1.1.3 Update items

ID	UCA03
Description	The user updates the detail information about the items.
Precondition	The user logged in with a seller account and has added at least one item into the system.
Basic flow	<ol style="list-style-type: none"> 1. The user chooses to search all the items in the database. 2. The system returns a list of the items which are added by the user in the database. 3. The user selects one item to edit in the list by clicking the "Update" button. 4. The system directs to a new page to show the detail information about the item. 5. The user updates the information and clicks the save button to submit the changes. 6. The system receives the change and saves it into the database. 7. The system redirects to the home page.
Alternative flow	<ol style="list-style-type: none"> 5a. The user gives up to edit the item and clicks the cancel button. <ol style="list-style-type: none"> 5a1. The system redirects to the home page. 6a. The information filled in the form is incorrect and the system alerts the user with a message. <ol style="list-style-type: none"> 6a1. The system keeps in the edit page waiting for the user to correct the information.
Post condition	The detail information about the item is changed in the database.

1.1.4 Delete items

ID	UCA04
Description	The user deletes the item from the database.
Precondition	The user logged in with a seller account and has added at least

	one item into the system.
Basic flow	<ol style="list-style-type: none"> 1. The user chooses to search all the items in the database. 2. The system returns a list of the items which are added by the user in the database. 3. The user selects one item in the list by clicking the "Delete" button. 4. The system alerts the user with a confirm message. 5. The user confirm to delete the item by clicking the "Yes" button. 6. The system deletes the record of that item from the database. 7. The system redirects to the home page.
Alternative flow	<ol style="list-style-type: none"> 5a. The user gives up to delete the item by clicking the "No" button. 5a1. The system returns to the home page.
Post condition	The item is deleted from the system.

1.2 Use case for feature B

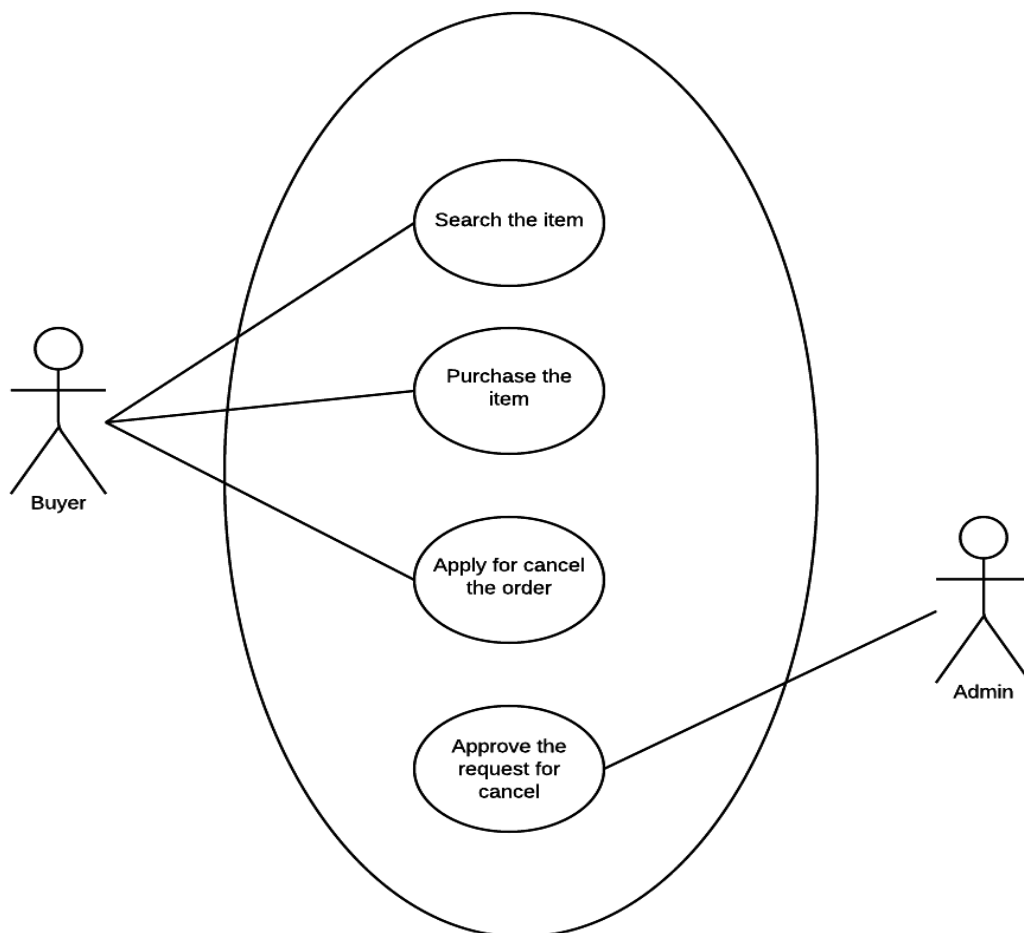


Figure 1.2 Use case diagram for feature B

1.2.1 Search items

ID	UCB01
Description	The buyer can search the items by entering a searching string.
Precondition	The user logged in with a buyer account
Basic flow	<ol style="list-style-type: none">1. The user enters a keyword to the system to search the certain type of products.2. The system returns a list of products related to the entering string.3. The user select items in the list to check the details by clicking the name of the item.4. The system returns a page of the information about the item.
Alternative flow	None
Post condition	None

1.2.2 Purchase items

ID	UCB02
Description	The buyer can purchase one or more items.
Precondition	The user logged in with a buyer account and selected one item to check the details.
Basic flow	<ol style="list-style-type: none">1. The user selects the number of the item to buy.2. The user clicks purchase button to buy the product.3. The system puts the products into the shopping cart to keep the shopping information for the user.4. The user confirms the details about the number and price of the trade and confirms the order.5. The system creates an order.
Alternative flow	<ol style="list-style-type: none">2a. The user presses cancel button to give up shopping.<ol style="list-style-type: none">2a1. The system returns to the item list page.4a. The user doubts about the details and cancels the trade.<ol style="list-style-type: none">4a1. The system removes the items from the shopping cart and redirects to the item list.
Post condition	An order is generated.

1.2.3 Apply for cancelling the order

ID	UCB03
Description	The buyer can apply for cancelling an order.
Precondition	The user logged in with a buyer account, and an order is created by the system with the user account.
Basic flow	<ol style="list-style-type: none">1. The user selects the order to apply for cancelling.2. The system sets a request for the order cancellation to the seller.
Alternative flow	None
Post condition	A cancellation request is generated.

1.2.4 Approve the request for order cancellation

ID	UCB04
Description	The seller can approve the request for order cancellation.
Precondition	The user logged in with a seller account, and a order cancellation request is sent to the user.

Basic flow	<ol style="list-style-type: none"> 1. The user clicks the "check request" button. 2. The system returns a list of the cancellation requests. 3. The user selects one request to confirm the cancellation by clicking the "confirm" button. 4. The system updates the order status and refreshes the order list.
Alternative flow	None
Post condition	The order is cancelled.

2. Domain class diagram

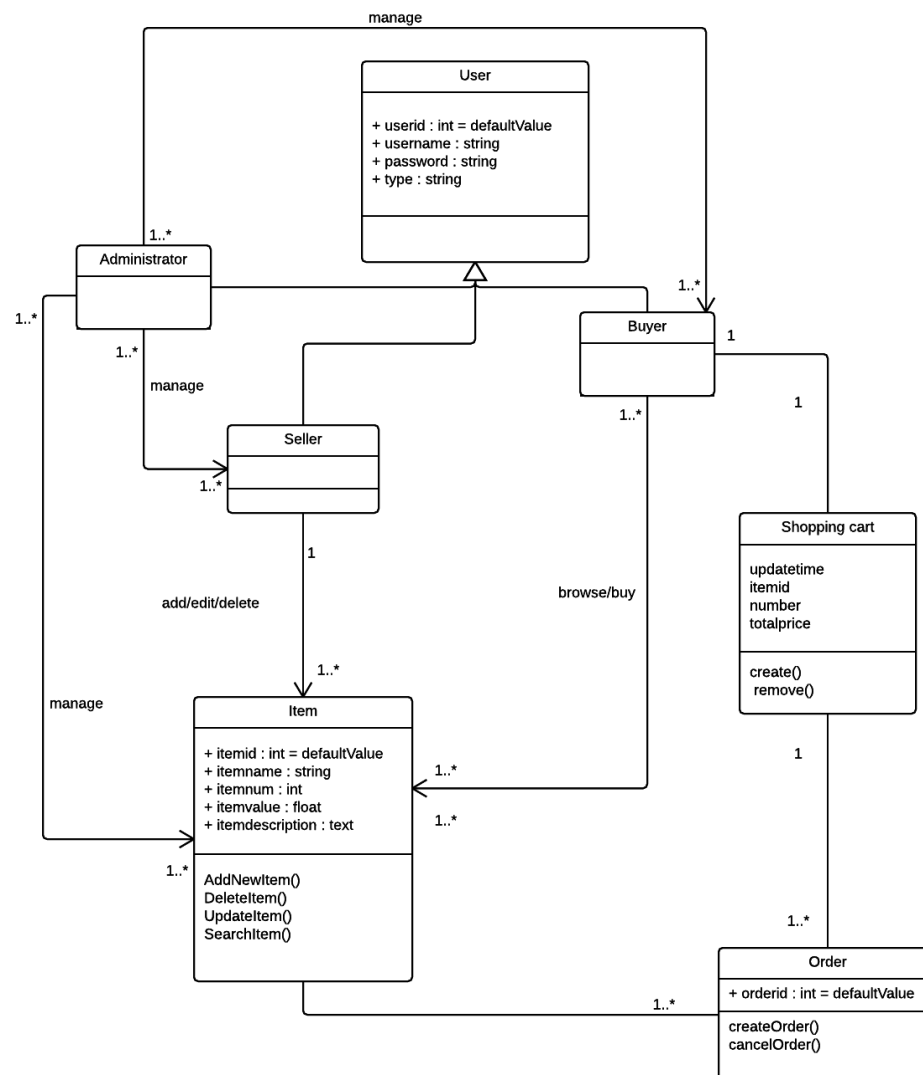


Figure 2.1 Domain class diagram

The domain class diagram for feature A and feature B contains the following classes:

User class: The user class is designed for all the clients, and it is the super class for Administrator class, Seller class and Buyer class.

Administrator class: The administrator class contains the attributes for the administrator user type, and the methods for administrator users to manage other user type accounts.

Seller class: The seller class is used for generating the objects of seller users. It has the attributes and methods of the seller user type features. Item management and order management are also related to this class.

Buyer class: The buyer class is for the buyer user type. It includes the attributes and methods of create new buyer accounts and other profile management.

Item class: The item class is used for item management. It contains the attributes for the products. For the methods, it has the methods about the item object creating, updating as well as deleting.

Shopping cart class: The shopping cart object is designed to store the product information temporary.

Order class: The order class is an important part of the system. It contains the attributes about the trade status and information such as the product number and total price. It has the methods about the order generating and cancelling.

3.Architecture design

3.1 High level architecture

Online shopping system is a layered system, and its architecture is as the image below.

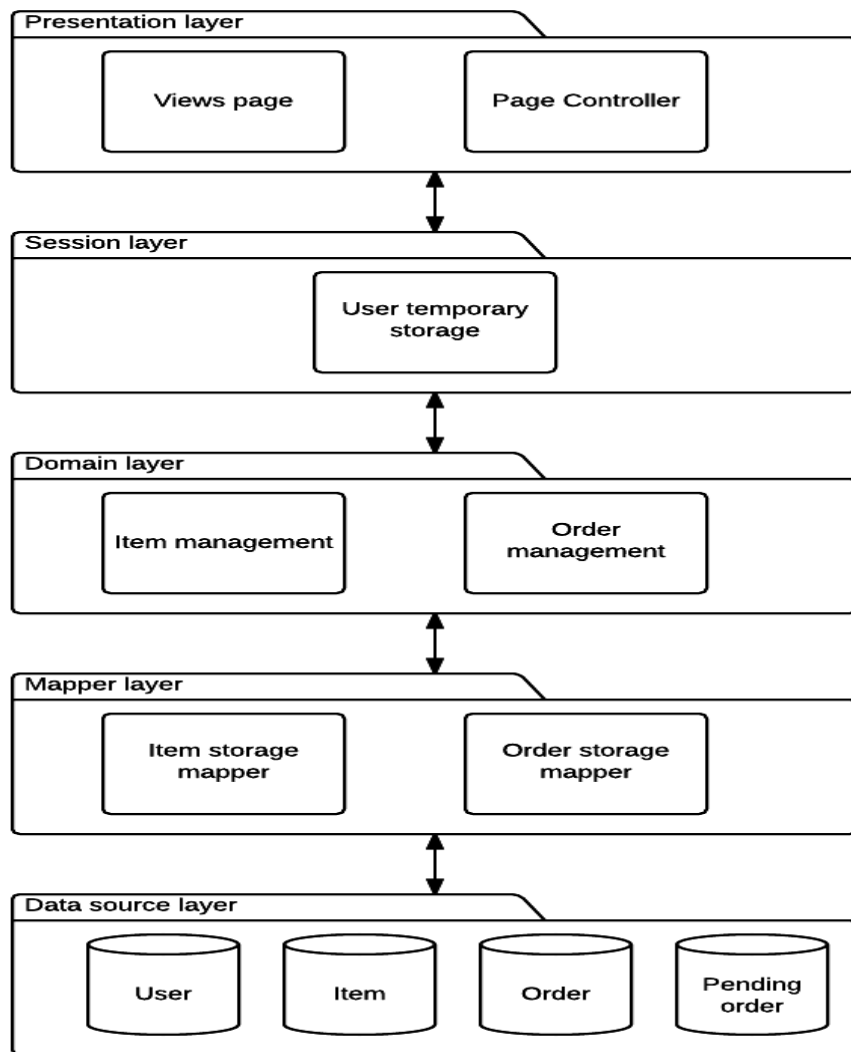


Image 3-1 High level architecture

And the responsibility of each layer is as below

Layer	Responsibility
Presentation	<ul style="list-style-type: none"> Transform the data it gets to a html page Display the information Response the request from clients with right service
Session	<ul style="list-style-type: none"> Store temporary data as login session
Domain logic	<ul style="list-style-type: none"> Provide trading logic service
Data mapper	<ul style="list-style-type: none"> Provides the mapping between domain objects and the entities in the data source layer
Data source	<ul style="list-style-type: none"> Data management and relevant logic rules for data

3.2 Presentation layer change for Feature B

As the feature B is mainly about ordering and order management and it can be separated from item management, what we need to do implement in presentation layer is adding extra necessary pages and their controller. Its contracture is changed as below.

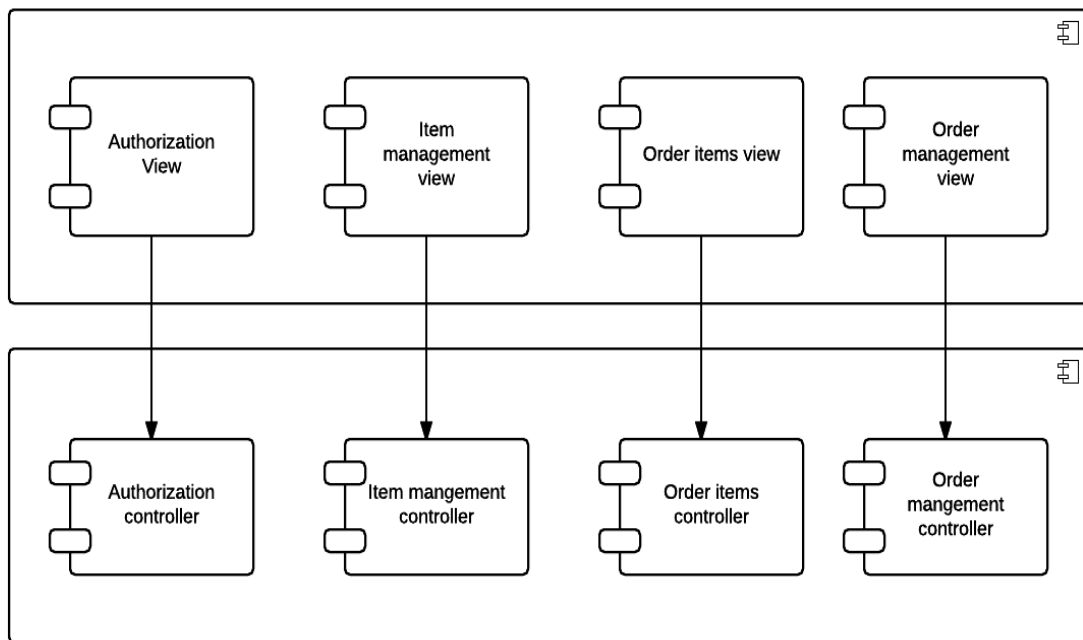


Image 3-2 Presentation layer

3.3 Session layer

As we need to check the status of logged in client and implement a shopping trolley to store what the client wants to buy, we store the message in the session. It is reasonable to store the information locally to avoid unnecessary communications with the server, because frequently communication will slow down the whole system, especially lots of user are using the system concurrently. And the structure for session layer is as the image below.

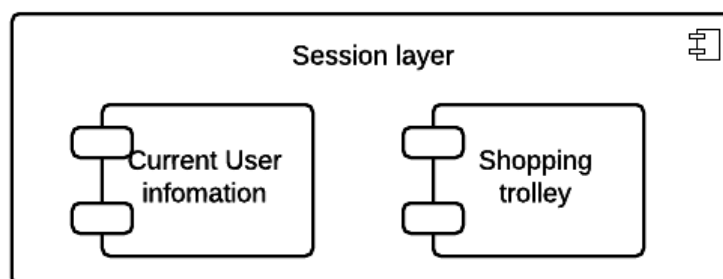


Image 3-3 Session layer

3.4 Domain layer for feature B

3.4.1 Layer structure

As we have session, we don't need to implement feature B with the help from authorization part. So the domain logic of feature B can be developed separately. The changed domain layer structure is as the image below.

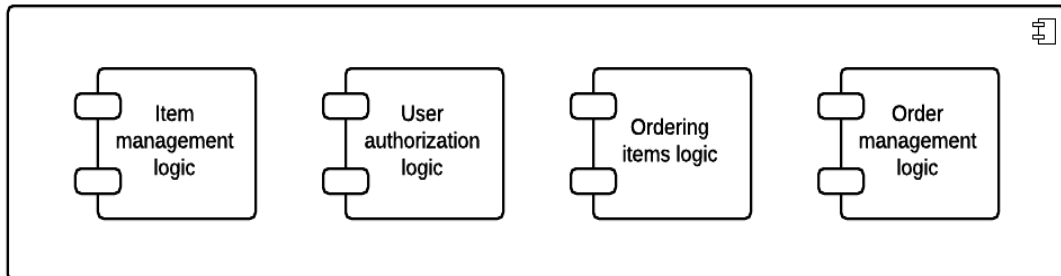


Image 3-4 Domain Layer

3.4.2 Layer behavior

The new behaviors for ordering and order management is as below:

Interface	Description
search (keyword:string)	Call the data mapper layer to go through the item table and list all items have the keyword in their name
buy (params)	Add the selected item to the shopping trolley
trolley ()	Call the data mapper layer to get the detail information for every item in the session
delete_item_in_trolley(id:int)	Delete one of the items from the session
trolley_purchase ()	Collection selected items info and call the data mapper to create an order for these items
my_orders ()	Call the data mapper to get current user's orders
ask_cancel (id:int)	Call the data mapper to change an order to a pending order
pending_orders ()	Call the data mapper to get all current user's pending orders
order_management ()	Call the data mapper to get all pending orders for admin
approve_cancel (id:int)	Call the data mapper to delete the pending order

3.5 Data mapper layer for feature B

3.5.1 Layer structure

We need to change the input from domain layer to the format stored in the database in this layer. As the most of logic is separated from the feature A, we will have two new mappers. The architecture is as below:

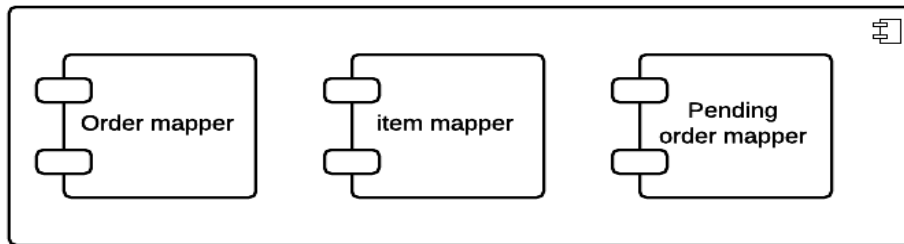


Image 3-5 Data mapper layer

3.5.2 Layer behavior

Interface	Description
search (keyword:string)	Go through the item table and return the result list
create_order (params)	Check the database and create an order with the parameters
list_order ()	Go through the order table and return current user's orders
createPendingOrder (oldOrder:order)	Create a pending order based on the input order and delete the old order from order table
pendingOrder (id:int)	Go through the pending order table and return current user's pending orders
all_orders ()	Return all pending orders for admin
approve (id:int)	Delete the pending order and update the item database for relevant items

3.6 Data source layer for feature B

We create two new tables to store orders and pending orders and add a new attribute to user table to determine whether he or she is an administrator.

4. Design pattern choice and justification

4.1 Presentation Layer

The presentation layer is used as the interface with the user, it will accept the requests and present the results to the user. It consists of two parts, controllers and views.

4.1.1 Controller

Page controller design pattern is used in the shopping online system. For the page controller design pattern, each page has its own controller, and the actions which are related to the business logic in this page are managed by this corresponding controller.

Even the front controller can reduce the code duplication, the page controller is more straightforward and simpler for the developers to implement. It also cost people who are unacquainted to the system less time to understand it and maintain it. For the feature B, purchasing items and order management, the controller design is not changed. The new business logic actions are clear and they can be added easily according to the rules of the page controller pattern, in other words, the number of the new controller is as same as the number of pages created for feature B.

4.1.2 View

Template view pattern is selected for the shopping online system. The template view pattern compacts with the page controller pattern very well. They both have the advantages of straightforward and simplicity. The emphasis for the system design should be the domain logic and data source layers, on the contrary, the presentation layer should be simple enough. For this reason, the transform view pattern and two step view pattern are too complex for the shopping online system we are implementing. For the feature B, the template view pattern is adapted for the same reason.

4.2 Domain Logic Layer

The domain logic layer is used for enforcing the business rules and managing the domain logic. Domain model pattern is chosen for the domain logic layer in the shopping online system. It is an object-oriented pattern which has the advantages of good extensibility and re-usability. For the feature B, we add new business rules to the system, in other words, the business rules of the system are changed. The domain model pattern is designed and used to solve this problem by modifying the modules. We can add new features to the system easily based on the feature A since adapting the domain model pattern.

4.3 Data Source Layer

The data source layer is used to store data. It also provides services to the domain logic layer. Three types of design should be considered: architectural pattern, behavioral pattern, and structural patter.

4.3.1 Architectural design

For the architectural design, data mapper patter is selected as the design pattern. It is an intermediate of the transfer between the database and the domain logic layer,

in order to prevent the domain layer to know the schema of the database to achieve the de-coupling. It can also improve the re-usability when combining with the domain model pattern.

4.3.2 Behavioral design

For the architectural design, data mapper pattern is selected as the design pattern. It is an intermediate of the transfer between the database and the domain logic layer, in order to prevent the domain layer to know the schema of the database to achieve the de-coupling. It can also improve the re-usability when combining with the domain model pattern.

4.3.3 Structural design

Foreign key mapping pattern is used in the structural design. The relations of the database tables reflect the relationships of the domain objects, which also can be used for new features.

4.4 Sessions

Client session state pattern is used in the system design. For the shopping online system, the session is used in the shopping cart part. After purchasing items, the session state is stored on the client. As to our system design, the items put in the shopping chart should be paid or the order should be created as soon as possible. We do not want to let the buyer keep the information of the item for long period. In this way, the client session state pattern is chosen. Compare to the server session state and database session state patterns, it can reduce the complexity of the server and database design, and improve the performance of the system server.

4.5 Concurrency

Optimistic offline lock pattern is used in the shopping online system. If two clients select the last item to purchase, while the order creating period, if one client has finished the action, the transaction of the other client will be roll back. The chance of this kind of action is low and using the optimistic lock can solve the problem appropriately with good liveness.

4.6 Security

Authentication enforcer pattern and authorization enforcer pattern are used in the system. For the authentication enforcer pattern, we use the username and password to implement. Using this pattern is to verify who the person is, and it will improve

the security and maintainability. As to the authorization enforcer pattern, we have different roles in the shopping online system, so this pattern is necessary as it claims that what the person can do with the system.

5. Dynamic behavior

5.1 Scenario one: User searches items

The user can search items by entering a string as the keyword, and then searching for all items related to that keyword.

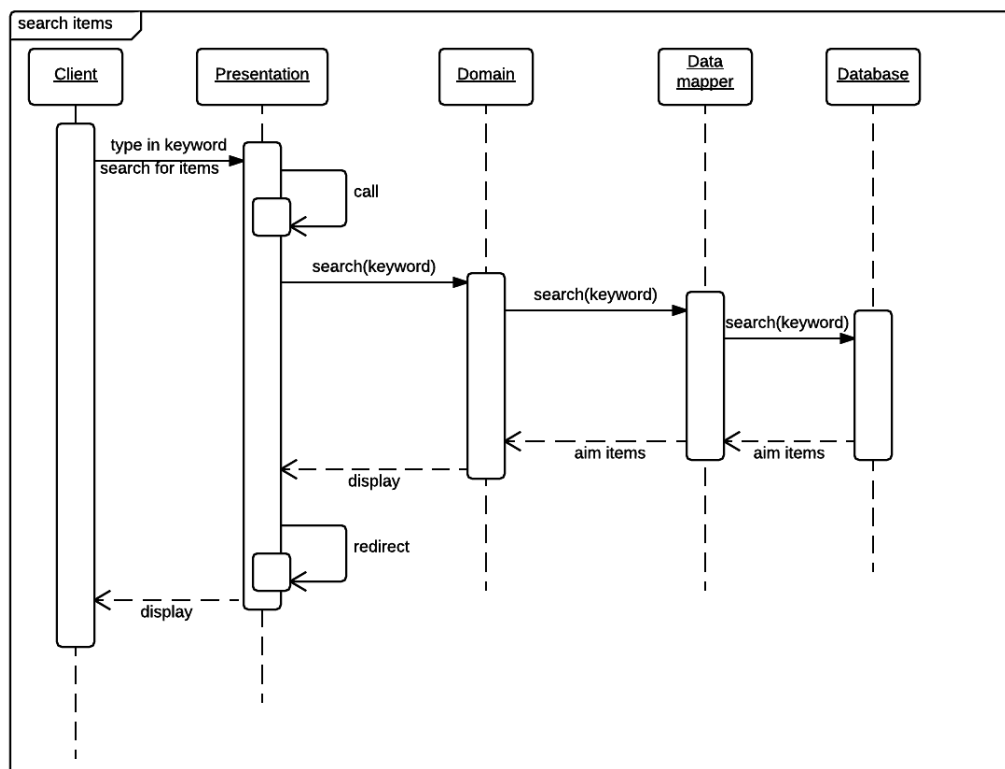


Image 5-1 Search sequence diagram

5.2 Scenario two: Buyer purchases items

The buyer can select one or more items to purchase. Firstly, the buyer selects item and quantity of the item, and puts the items into the shopping chart. Secondly, after confirming the items details, the order can be created.

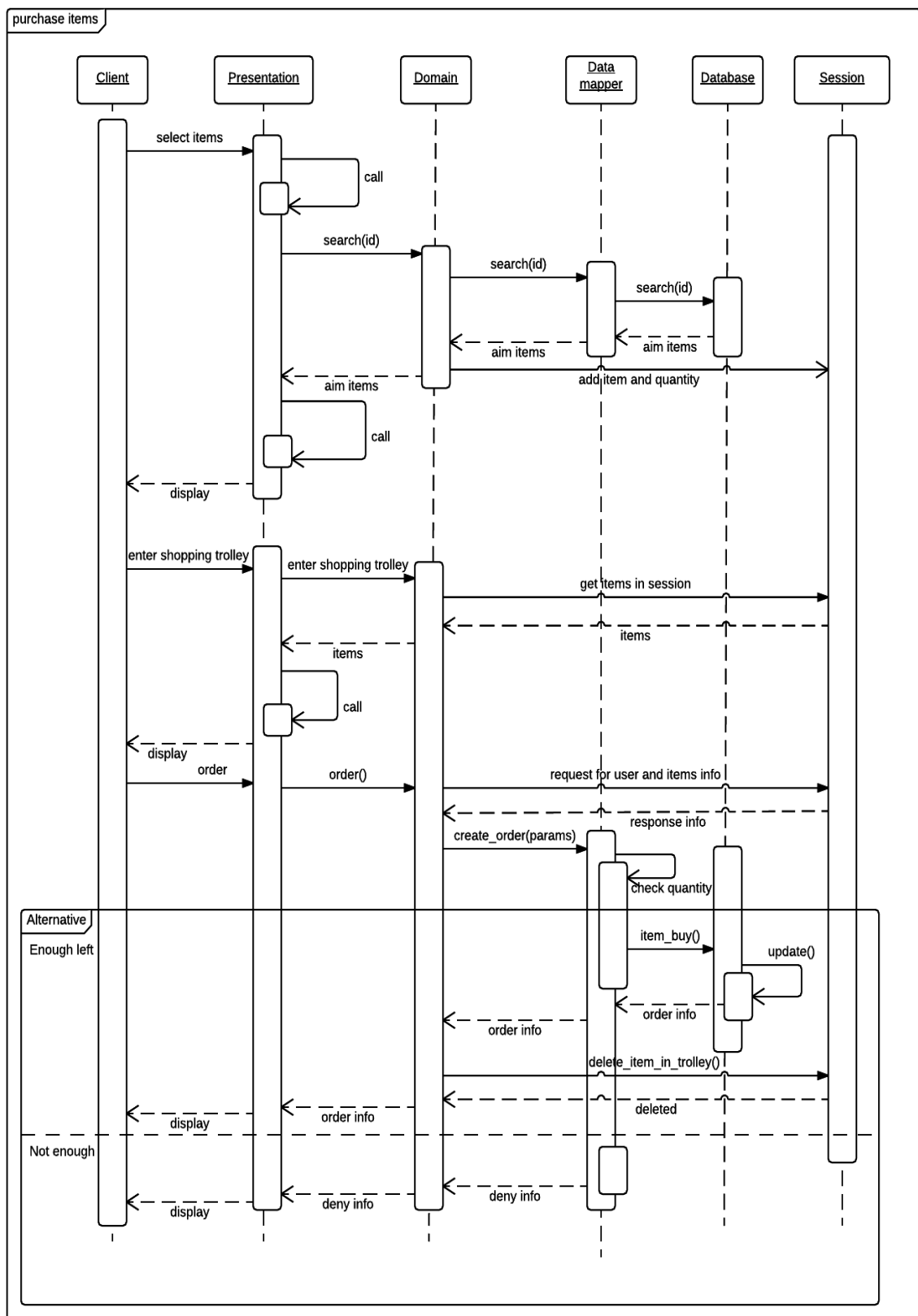


Image 5-2 Purchase items sequence diagram

5.3 Scenario three: Buyer applies to cancel the order

The buyer can select the order to cancel the order after the order was created.

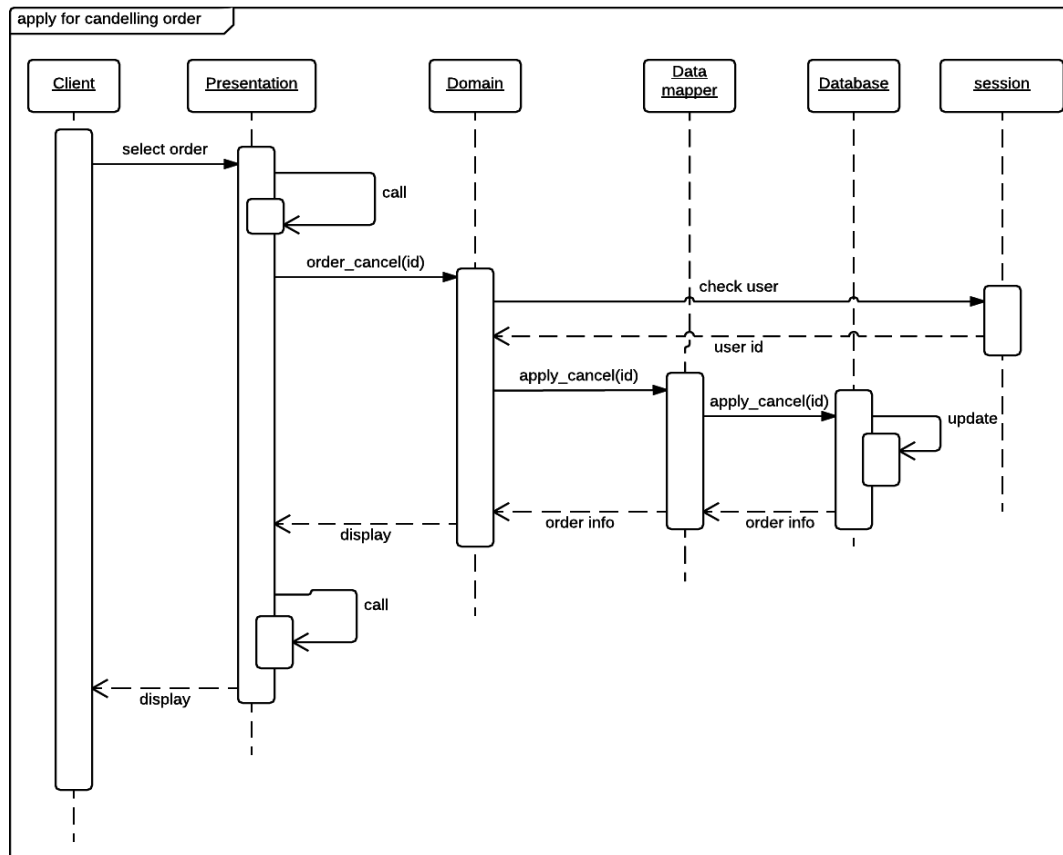


Image 5-3 Cancel an order sequence diagram

5.4 Scenario four: Admin approve the cancel request

Admin can approve the cancel request sent by the buyer.

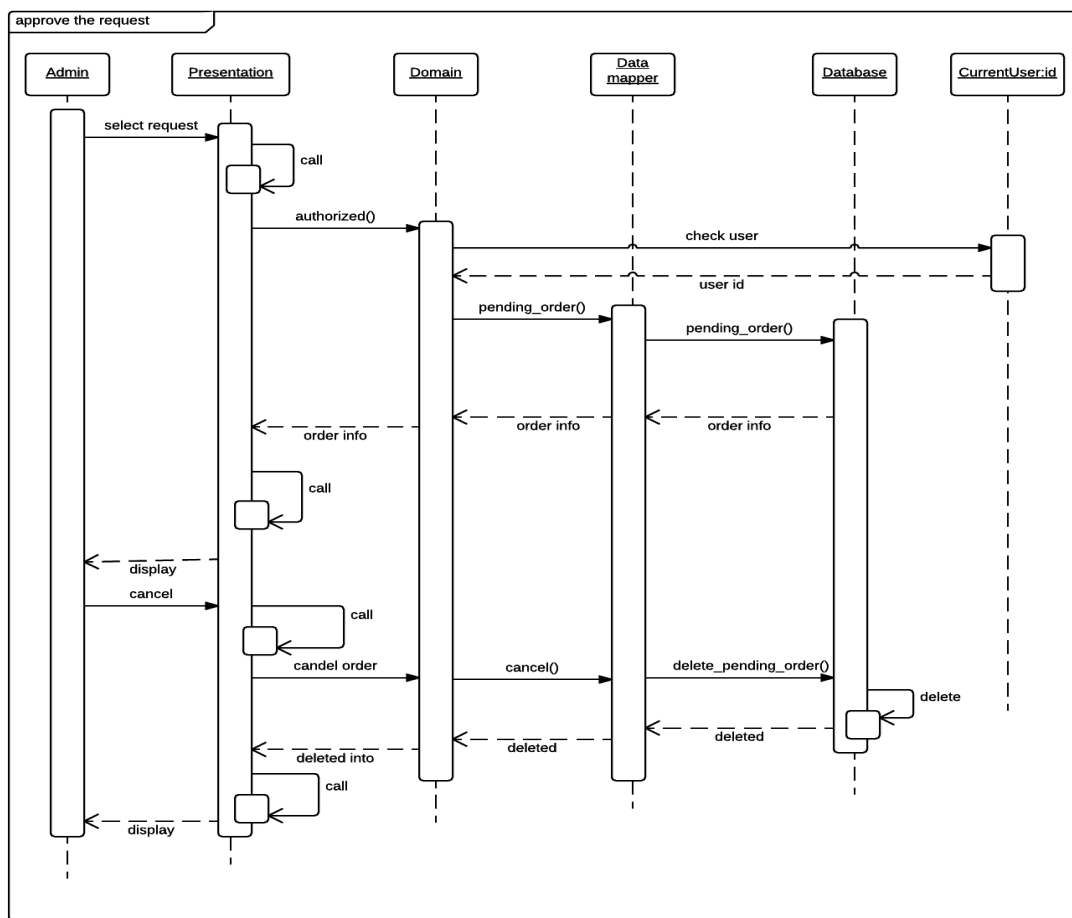


Image 5-4 Approve cancel sequence diagram