Submission 2. Architecture design for item management

Group 28
Keran Wang (keranw 686976)
Xue Jiang (jiangx2 665401)

1.Introduction

The enterprise system we are working on is an online shopping system, and the feature we will design and implement in this submission is “item management” for item sellers.

2.Domain Class Diagram

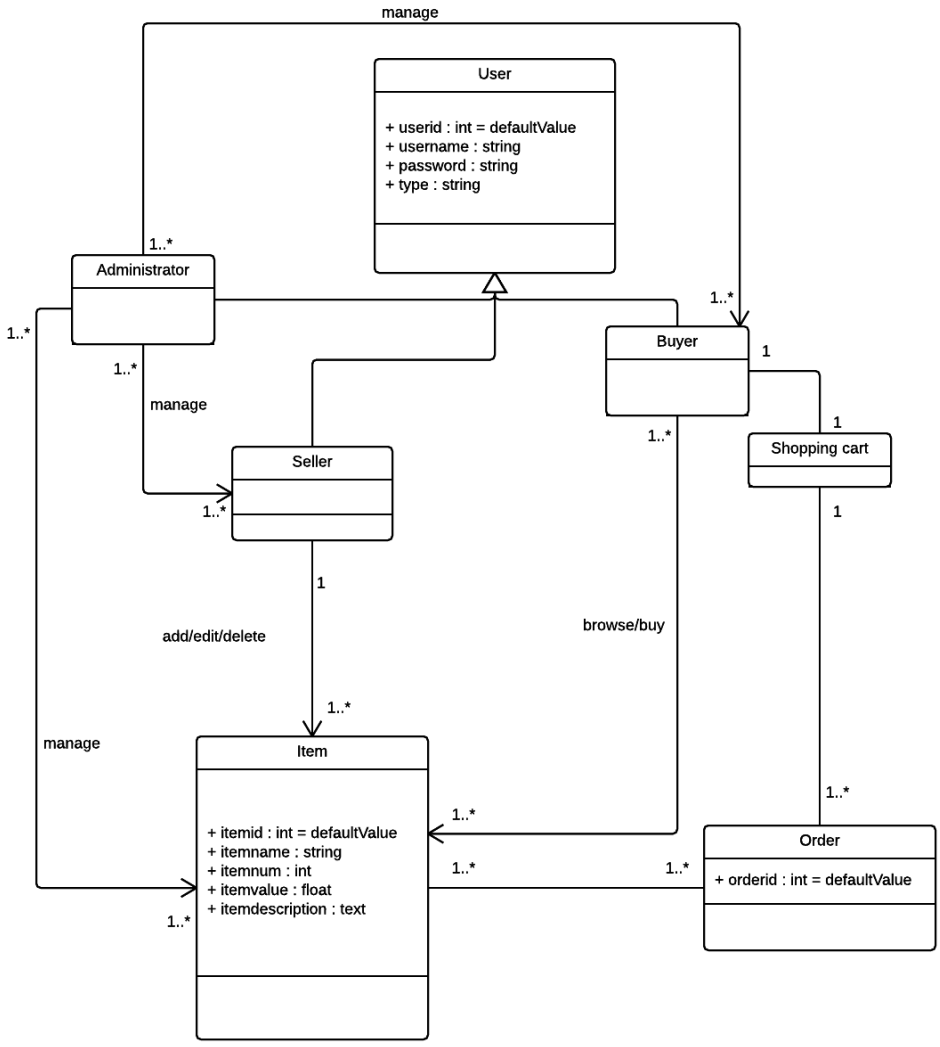


Figure 2.1 Domain class diagram

The domain class model consists of the following classes:

- **User class**: The clients for the online shopping system, it has three sub classes.
    - **Administrator class**: The account for the employee from the online shopping system company. This kind of account can manage the other types of accounts and other objects information in the system.
    - **Seller class**: The account for the client who has registered as a seller in the system. The seller can manage the items and the orders.
    - **Buyer class**: The account for the ordinary user who is willing to buy some items through the system. This class stores the information about the user.
- **Item class**: The goods that selling through the online shopping system. The seller takes charge of his own items.
- **Shopping cart class**: The buyer selects an item which he wants to buy. The item is put in the shopping cart waiting to be paid, which is a process of commodity transaction.
- **Order class**: After the buyer paid for the item, an order is generated. It contains the information about the transaction.

3.Architecture Design

3.1 High Level Architecture

Shopping online system is a layered system, and the high-level architectural diagram is shown in Figure 3.1.

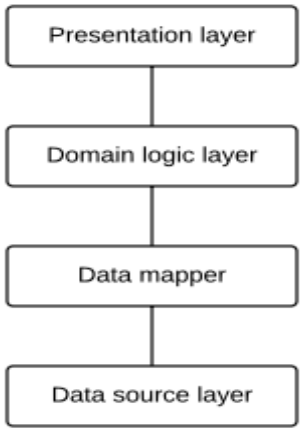


Figure 3.1 High-level architectural diagram

The responsibilities are as below.

Layer	Responsibility
Presentation	• Transform the data it gets to a html page • Display the information • Response the request from clients with right service
Domain logic	• Provide trading logic service
Data mapper	• Provides the mapping between domain objects and the entities in the data source layer
Data source	• Data management and relevant logic rules for data

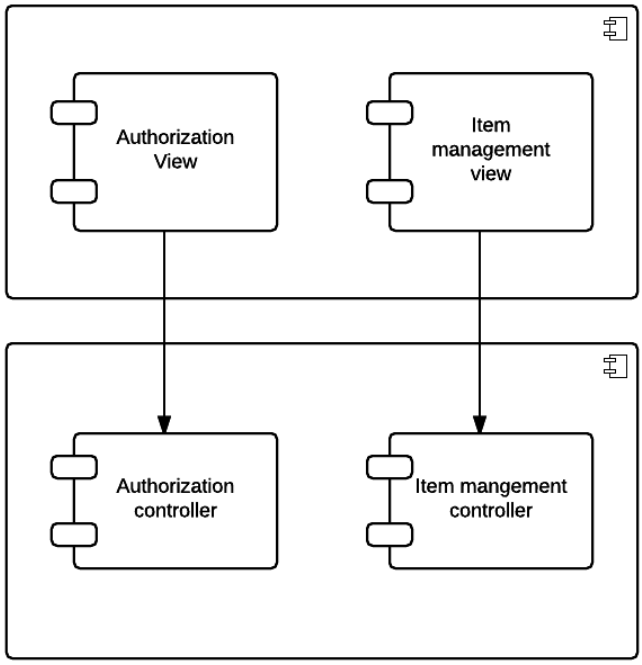
3.2 Presentation Layer

3.2.1 Layer description

The presentation layer is responsible for transferring requests to the domain logic layer from the user interfaces, and returning the data and services back to the user. It provides interfaces and web pages for users in order to collect data and requests, and the data or requests were received by the domain logic layer. The presentation layer contains views and controllers, views create different view templates to interact with the users, and controllers manage requests from the users. In this system, the page controller pattern and the template view pattern are used.

3.2.2 Layer structure

This layer has page views and direct controllers. The controllers deal with requests from clients and every page has a specific controller. The views generate pages with data they get from domain layer service their controllers show.



Feature 3.2.1 Presentation layer structure

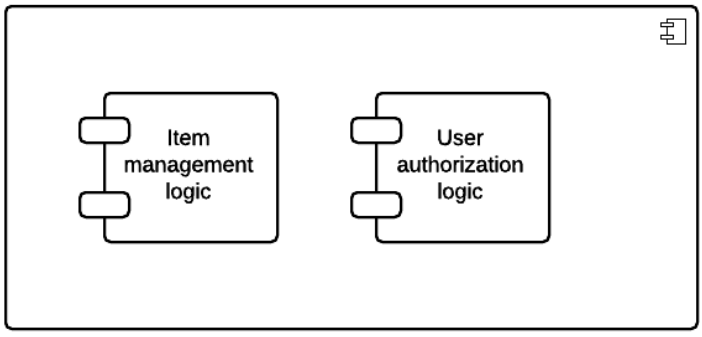
3.3 Domain Layer

3.3.1 Layer description

The domain logic layer is the layer which the business rules are implemented in. The responsibility of the domain logic layer is that it responds and handles the user requests sent from the presentation layer, and transforms and processes data with data mapper. In the online shopping system, domain model pattern is used in the Domain logic layer.

3.3.1 Layer structure

This layer for an online shopping system should contain different domain logic for online trading. The feature we choose to implement this time is item management, and all the domain logic is relevant with the object “item”.



Feature 3.3.1 Domain layer structure

3.3.2 Layer behaviour

The module which describe the domain logic of item management has 5 specific interfaces.

Interface	Description
Item_query (owner_id:int)	Call the data mapper layer to get the items belong to current user for statistic, and the returned array is an array of the items information
Item_show (id:int)	Call the data mapper layer to get the item information with the specific item id, and the returned object is the item
Item_create (params)	Call the data mapper layer to add a new item for the user to sell with the parameters the user supports
Item_edit (params)	Call the data mapper layer to update the information of a specific item with the parameters the user supports
Item_destroy (id:int)	Call the data mapper layer to delete an item that the user doesn't want to sale anymore with the id of the item

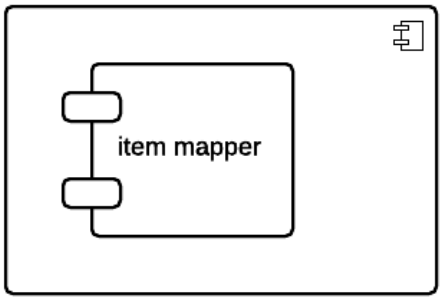
3.4 Data Mapper Layer

3.4.1 Layer description

The data mapper layer is used as the data mapper pattern which separate the domain object from the database. It takes the responsibility of handling data result sets from the database and transforms it with the domain object. The data mapper pattern decreases the coupling between the database and domain object.

3.4.1 Layer structure

We will have a data mapper layer to move the data from a domain object to the database for all objects in this system to keep domain objects and database independent. The structure for item management in this layer is as the image below.



Feature 2.4.1 Data mapper layer structure

3.4.2 Layer behavior

The interfaces for item managements in this layer are as below.

Interface	Description
query_with_owner_id	User the owner's id (the foreign key of item) to find the his or her

(owner_id:int)	items and push the information in an array, and return the array to domain layer
query_with_item_id (id:int)	User the id of the item to find its information from the database, and return the information to domain layer
Create_an_item (params)	Receive the parameters from the domain layer, maps the data and insert the mapped data to the database
delete_an_item (id:int)	Delete the item from the database with its id
edit_an_item (params)	Receive the parameters from the domain layer, maps the data and update the data of that item

3.5 Data Source Layer

3.5.1 Layer description

The data source layer is responsible for storing the data and processing the queries from the data mapper. Entities and relationships are stored in the database. In the system, lazy load pattern and foreign key mapping are used.

3.5.2 Layer structure

This is the layer change the database directly. The prototype needs at least two tables to record the item information and user information, and we have an attribute in item table called “owner_id” as the foreign key to record the relationship between the two tables.

4. Design pattern choice and justification

4.1 Presentation Layer

The presentation layer is responsible for providing the user interfaces, processing the user requests, and presenting the outcome to the user.

The presentation layer is comprised of controllers and views. In the shopping online system, page controller pattern is chosen for the input controller, and template view pattern is selected as the view pattern.

4.1.1 Controller

Page controller design pattern is described as one controller takes charge of one page or one action. It deals with the business logic related to user actions.

Another alternative option for controller pattern choice is front controller, which is not used in the shopping online system. For the front controller pattern, one controller handles all the web pages and requests.

Compare to the front controller, the page controller is much more straightforward than the other one. Using the front controller pattern can solve the problem of logic and code duplication by allocating all the requests in one controller, but the page controller is easier to understand and is simpler to implement.

Besides that, although the online shopping system is an enterprise application with complex business rules, the number of view templates is moderate, which means that the disadvantage of code duplication for page controller is more acceptable than the design complexity by using the front controller.

In addition, for developers who are unfamiliar with the system but have to maintain it, keeping controllers separated by using page controller pattern will help them understand the system better and find faults with less time.

4.1.2 View

Template view pattern is generating the view with embedded dynamic information in the static web page. The biggest advantage for the template view pattern is simplicity, and combining it with the page controller, it will be more straightforward than any other view pattern.

For the other two patterns, the transform view and two step view, they all have the shortcoming of high complexity including design complexity and implementation complexity. For the shopping online system, the mainly design points should focus on the domain logic layer and data source layer rather than the presentation layer, in other words, the presentation layer design should be as simple as possible.

4.2 Domain Logic Layer

The domain logic layer is responsible for implementing the business rules and organizing the domain logic.

In the shopping online system, domain model is selected as the domain logic layer pattern. The domain model is an object-oriented pattern which allocates the behaviour to the objects based on the business logic by using methods. It decreases the coupling and increase the re-usability.

The domain model pattern has good extensibility, therefore, if the business rules of the system are changing, the modules can be modified accordingly. In this respect, the shopping online system has complex business rules which are subject to change, and the feature we implement, item management, is just a basic feature in the online shopping system, and there should be more features are object oriented. So we need a pattern that is easy to extend, in other words, the domain model suits the system.

Compare to the domain model, the other two pattern belong to the domain logic layer, transaction script and table module, are more suitable for the system which has simple domain logic. In addition, the transaction script is a procedure-oriented pattern which is not compatible with our online shopping system.

4.3 Data Source Layer

The data source layer is responsible for storing data and providing servers to the domain logic layer. The data source layer design includes three parts: architectural design, behavioural design and structural design. In the shopping online system, we adopt data mapper pattern for architectural design, lazy load pattern for behavioural design, and foreign key mapping for structural design.

4.3.1 Architectural design

Data mapper pattern is used for the data source layer architectural design. The data mapper pattern is acting as an intermediate between the domain object and the database. The domain layer does not know the database schema. In this way, the data mapper de-couples the domain logic layer and the database to prevent the data in the database. Compare with the row data gateway pattern and active record pattern, the data mapper pattern has loose coupling.

In addition, the data mapper is highly compatible with the domain model pattern which we choose for the domain logic layer design. This combination can improve the re-usability for the database and the domain logic layer.

4.3.2 Behavioural design

The purpose of using the lazy load pattern in the behavioural design is to instead of reading amount of data from the database, just only reading the data that the request needs.

The superiority of using the lazy load pattern is the efficiency, and for the online shopping system which many users may use concurrently, we need a pattern to improve its performance.

4.3.3 Structural design

For the database structural design, foreign key mapping pattern is selected as the design pattern. The foreign key mapping pattern allows the domain relationships and the database tables mechanically associated, which can keep the association of different objects.

5. Dynamic Behaviour

As we have four specific scenarios, we will describe the interaction between layers in the system for theses four scenarios as below.

5.1 Scenario one: User retrieves all the items

After sellers log in, the shopping online system, they can choose to retrieve all the items which were added or belonged to them.

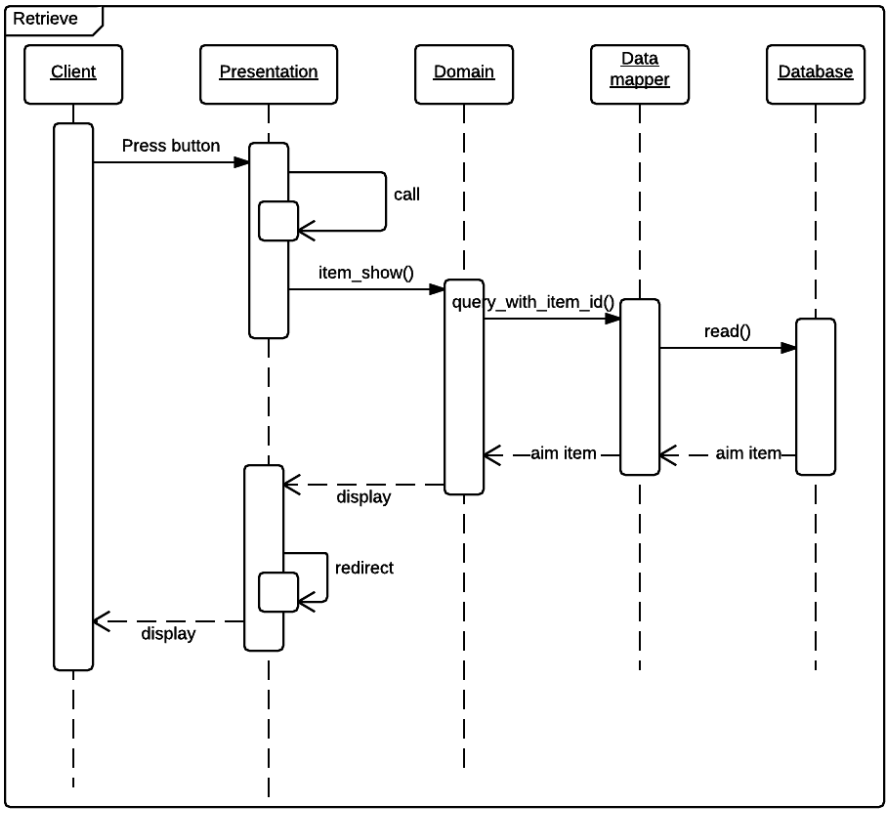


Figure 5.1 sequence diagram for item retrieve

5.2 Scenario two: User create new item

The seller users can add new items that they want to sell. After pressed the add new item button, the system will provide the user with a form table to fill. Then the seller input the information about the new item, and submit it to the system. At last, the system will store that item and show it in the item list.

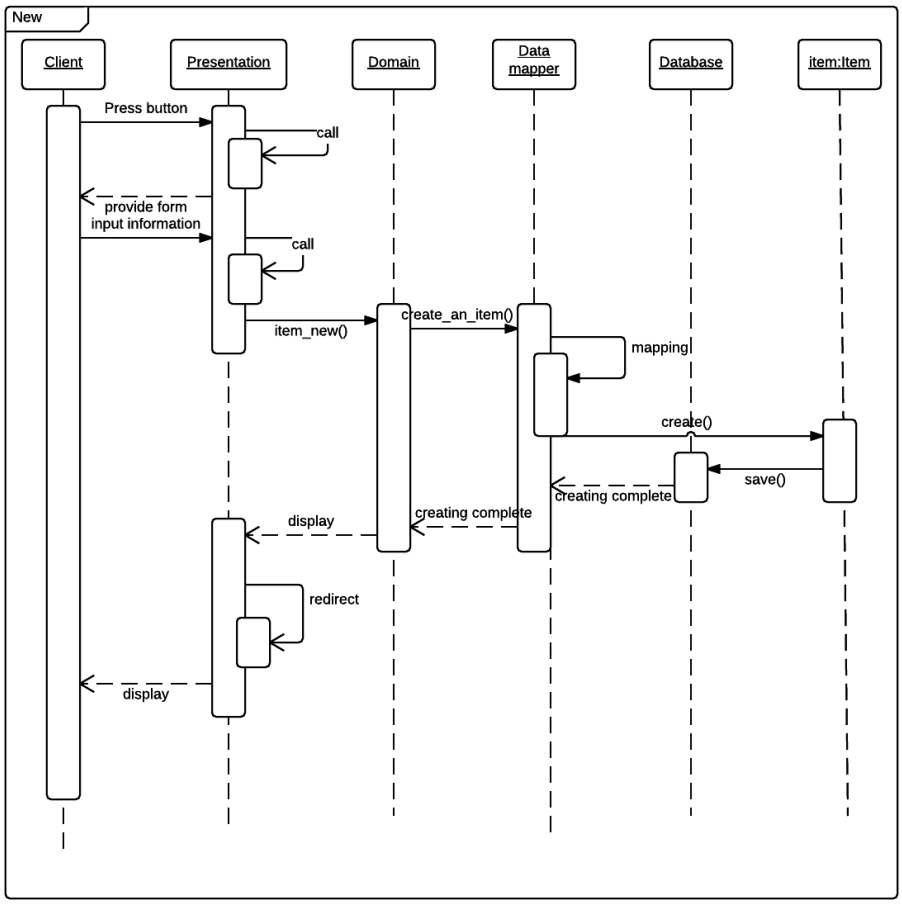


Figure 5.2 sequence diagram for create new item

5.3 Scenario three: User delete the item

The seller users can delete items that they do not want to sell any more. The user clicks the delete button, and the system will delete the item from the item list after receiving the confirmation from the user.

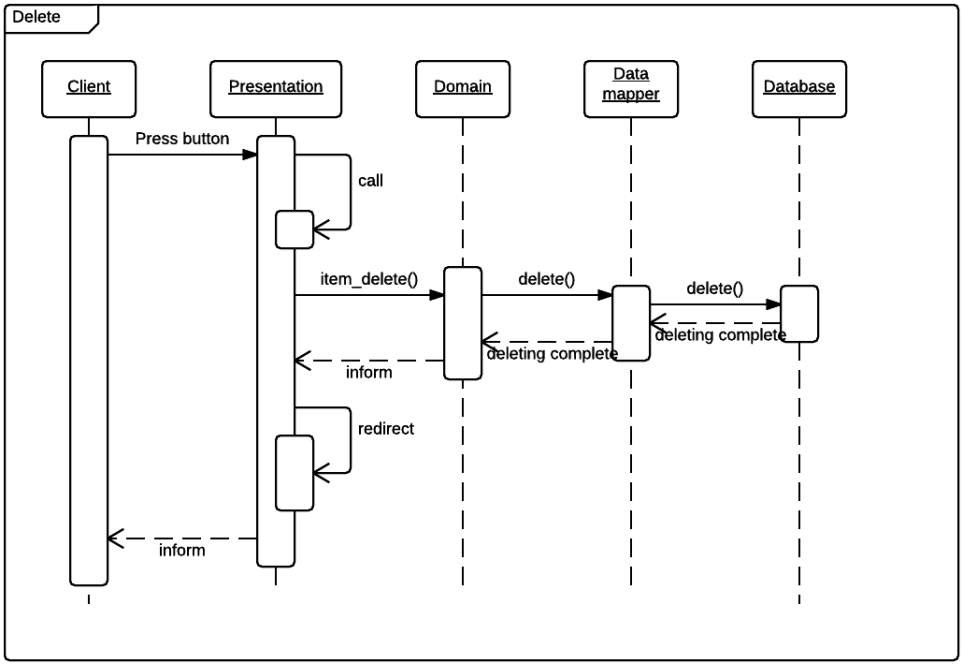


Figure 5.3 sequence diagram for item delete

5.4 Scenario four: User update the information about the item

The user can update the existing item details from the item list. The seller selects the item from the list, and the web will perform a table with the item detail. The user can edit the information in the table and submit it to system to save it. After that, the item details will be updated.

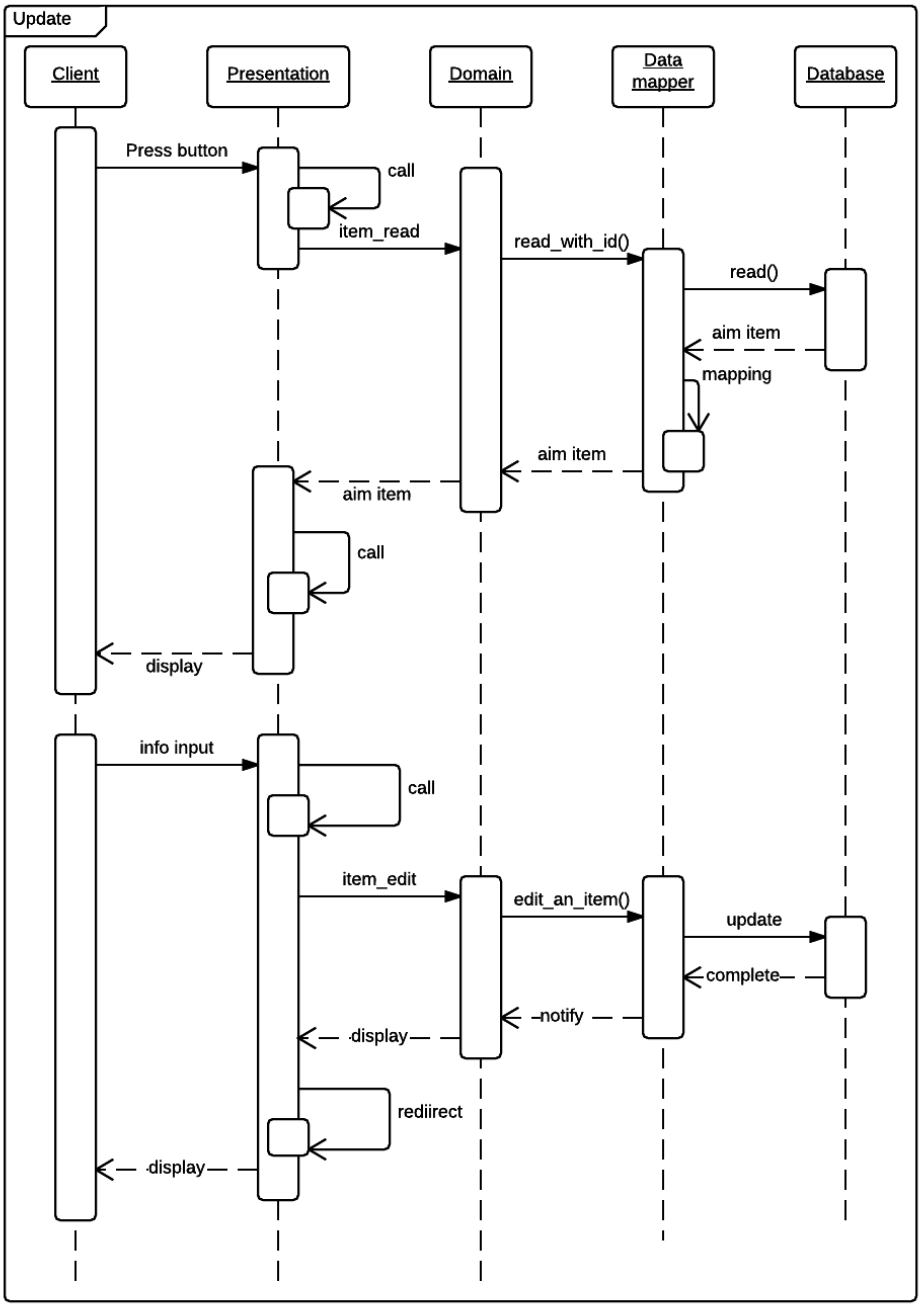


Figure 5.4 sequence diagram for update item information