OO Analysis and Design with UML and USDP

Solutions

Created by Dr. Jim Arlow

Version 2.0

Table of Contents

1 OOAD with UML and USDP - Sample Solutions

1.1 Introduction

This document provides a set of sample solutions to the Labs presented in the OO Analysis and Design with UML and USDP Workbook.

We have provided a sample set of documents that constitutes reasonably complete requirements, analysis and design models for the ECP along with the CRC card and noun/verb analysis artifacts.

Table 1 provides a mapping from each individual Lab to the appropriate part of this sample solution.

Table 1:

Workbook	Solutions
1 Requirements - Capturing requirements lab, page 1	
1.2 Functional requirements - 20 minutes, page 1	2 ECP System Requirements Specification v1.1, page 3
1.3 Non-functional requirements - 20 mins, page 3	
2 Requirements - Use case modelling lab, page 5	
2.2 Identifying actors - 20 mins, page 5	3 Use Case Model (Outlined), page 7
2.3 Identifying use cases - 40 mins, page 6	
2.4 Creating a use case diagram - 10 mins, page 7	
2.5 Detailing use cases - 40 mins, page 7	5 Use Case Model (Detailed), page 11
2.6 Creating a glossary - 10 mins, page 8	4 ECP Project Glossary version 1.1, page 10
3 Requirements - Advanced use case modelling lab, page 9	
3.2 Updating the use case diagram - 15 mins, page 9	5 Use Case Model (Detailed), page 11
3.3 Detailing the use cases - 15 mins, page 9	5 Use Case Model (Detailed), page 11
4 Analysis - Finding analysis classes lab, page 11	
4.2 CRC Brainstorm - 30 mins, page 11	5.5 CRC brainstorm, page 39
4.3 Noun/verb analysis - 20 mins, page 12	5.6 Noun/verb analysis, page 39
4.4 Consolidation - 20 mins, page 12	5.7 Consolidation, page 43
5 Analysis - Finding relationships lab, page 14	

5.2 Identifying associations - 20 mins, page 14	7 ECP analysis class diagrams, page 74
5.3 Other relationships - 20 mins, page 16	
6 Analysis - Use case realisation lab, page 17	
6.2 Creating communication diagrams - 15 mins, page 17	5.4 Use case detailed semantics, page 13
6.3 Creating sequence diagrams - 15 mins, page 18	
6.4 Finishing the realisations - 45 mins, page 18	
7 Design - The design model lab, page 21	
7.3 Subsystems and layers - 5 mins, page 22	8.2 Subsystems, page 78
7.4 Creating the design classes - 30 mins, page 22	8.3 Design class diagram, page 79
7.5 Sequence diagram - 15 mins, page 24	8.5 AddBookToCatalog design sequence diagram, page 80 8.6 DeleteBookFromCatalog design sequence diagram, page 81
8 Design - state machines lab, page 26	
8.2 Order processing - 30 mins, page 26	8.7 Order state machine, page 81
8.3 Adding and removing items from an order - 30 mins, page 26	8.8 UnpaidFor submachine state - adding and removing items, page 82
9 Deployment - Deployment and implementation lab, page 28	
9.2 Nodes - 15 mins, page 28	9.1 Deployment diagram, page 83
9.3 Components - 15 mins, page 29	9.1 Deployment diagram, page 83

2 ECP System Requirements Specification v1.1

ID	Details	Type	Priority
R1	The ECP shall display a list of all products offered by Clear View Training Limited.	• Products • Functional	MustHave
R2	The ECP shall organise the list of products by product category.	• Products • Functional	MustHave
R3	The ECP shall display detailed product descriptions consisting of name, photograph, price and description on demand.	• Products • Functional	MustHave
R4	The ECP shall display the number of items currently in the shopping basket on each page of the catalog.	• Products • Functional	Could-Have
R5	The ECP shall accept all major credit cards.	• Payment • Functional	MustHave
R6	The ECP shall validate payment with the credit card processing company.	• Payment • Functional	MustHave
R7	The ECP shall automatically calculate and add a shipping charge to the order.	• Payment • Functional	Should-Have
R8	The ECP shall automatically calculate and add tax to the order.	• Payment • Functional	Should-Have
R9	The ECP shall allow the customer to place items into a shopping basket.	• User Interface • Functional	MustHave
R10	The ECP shall allow the customer to remove items from their shopping basket.	• User Interface • Functional	MustHave
R11	The ECP shall allow the customer to “check out” and pay for their products.	• User Interface • Functional	MustHave
R12	The ECP user interface shall support the insertion of adverts.	• User Interface • Functional	WantTo-Have

R13	The ECP shall raise a new customer order at the point of payment.	• Orders • Functional	MustHave
R14	The ECP shall notify the dispatch department once the order has been paid for.	• Orders • Functional	Should-Have
R15	The ECP shall send a copy of the current order to the customer when payment is accepted and the order confirmed.	• Orders • Functional	MustHave
R16	The ECP shall allow customers to view their order history.	• Orders • Functional	Should-Have
R17	The ECP shall allow a customer to cancel a customer order provided that the order has not been dispatched.	• Orders • Functional	MustHave
R18	The ECP shall notify the dispatch department when an order is cancelled.	• Orders • Functional	MustHave
R19	When the order is cancelled the ECP shall refund the cost of the order.	• Orders • Functional	MustHave
R20	The ECP shall allow the dispatch department to view all orders.	• Orders • Functional	MustHave
R21	The ECP shall allow a customer to register with the site.	• Registration • Functional	MustHave
R22	The ECP shall use the customer's email address as the username for logon purposes.	• Registration • Functional	MustHave
R23	The ECP shall require the customer to set a password.	• Registration • Functional	MustHave
R24	The ECP shall collect customer information consisting of name, address, email address, phone number, fax number, credit card information.	• Registration • Functional	MustHave
R25	The ECP shall allow customers to view and edit their customer information.	• Registration • Functional	MustHave

R26	The ECP shall authenticate all customers prior to making payment, viewing outstanding orders or viewing customer information.	• Registration • Functional	MustHave
R27	The ECP shall support 100,000 transactions per day.	• Capacity • NonFunctional	Should-Have
R28	The ECP shall support a peak transaction rate of 10 transactions per second.	• Capacity • NonFunctional	Should-Have
R29	The ECP shall support 10,000 concurrent sessions.	• Capacity • NonFunctional	MustHave
R30	The ECP shall be available 24 hours per day, 360 days per year.	• Availability • NonFunctional	MustHave
R31	The ECP shall store sales transaction data.	• Availability • NonFunctional	MustHave
R32	The ECP shall accept payment and raise an order within 5 seconds in 95% of the cases. This figure might not always be achievable to the customer because of Internet latency and delays in payment authorisation by the credit card company.	• Performance • NonFunctional	Should-Have
R33	The ECP shall log in a customer within 5 seconds.	• Performance • NonFunctional	Should-Have
R34	The ECP shall use a browser as its user interface.	• ComplianceTo-Standards • NonFunctional	MustHave
R35	The ECP shall support the latest versions of Internet Explorer and Netscape browsers.	• ComplianceTo-Standards • NonFunctional	MustHave
R36	The ECP shall always support the previous version of Internet Explorer and Netscape browsers.	• ComplianceTo-Standards • NonFunctional	MustHave

R37	The ECP shall be written using standard Java to run on Linux and Windows.	• ComplianceTo-Standards • NonFunctional	Should-Have
R38	The ECP shall run on the same system and software as the existing Clear View Training web site.	• ComplianceTo-Standards • NonFunctional	MustHave
R39	The ECP shall authenticate all users of the system who are not customers.	• Security • NonFunctional	MustHave
R40	The ECP shall use secure means of communi-cation.	• Security • NonFunctional	MustHave
R41	The ECP shall produce non-refutable evi-dence of transactions.	• Security • NonFunctional	MustHave
R42	The ECP shall provide a security audit trail.	• Security • NonFunctional	MustHave

3 Use Case Model (Outlined)

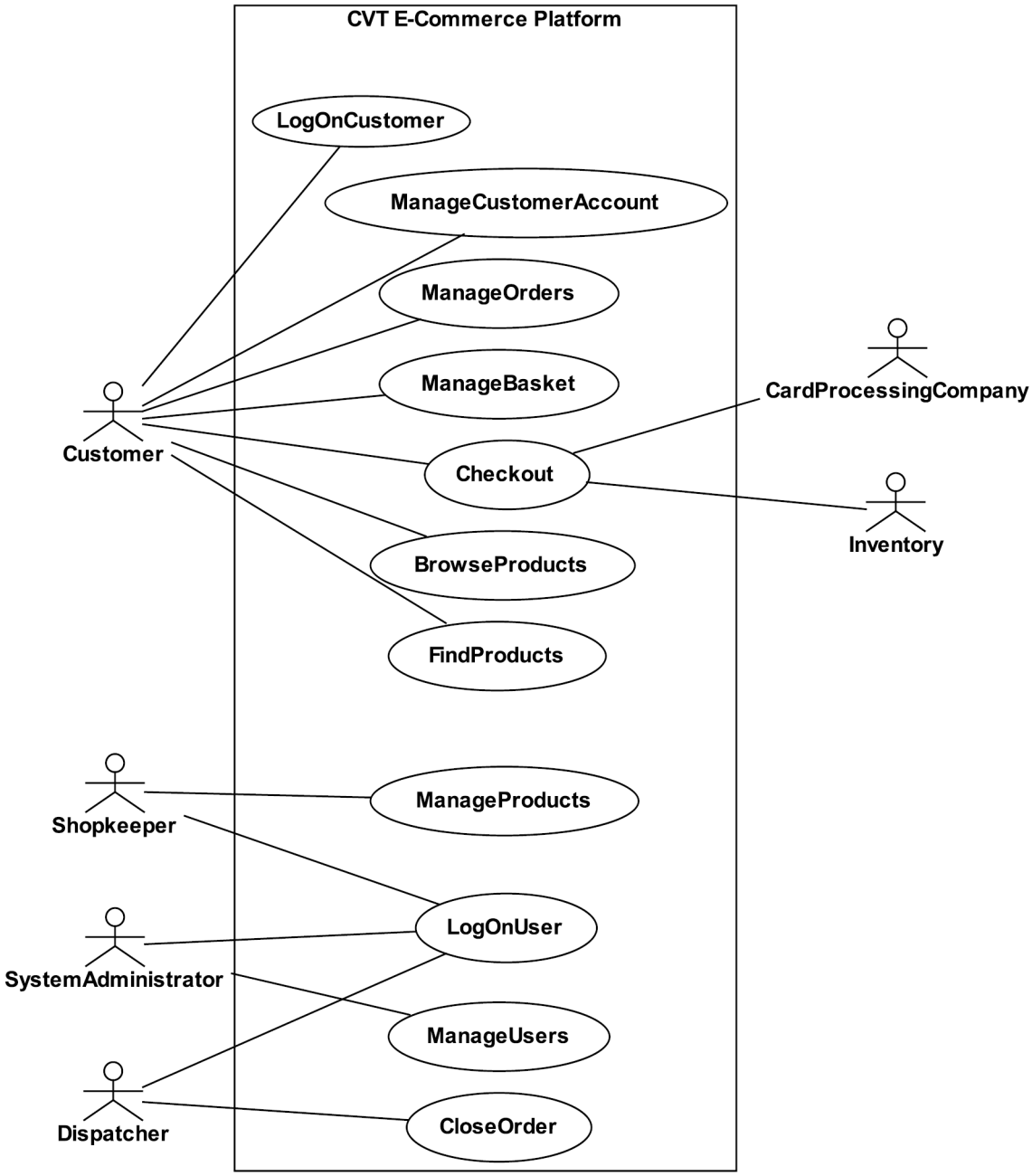
3.1 Introduction

This document contains the use case model (outlined) for the ECP.

3.2 Use case diagram (outlined)

This is a first-cut use case diagram for the ECP.

Figure 1:



3.3 Actor Semantics

The table below contains brief semantics for the actors in the ECP system.

Table 2:

Actor	Semantics

Table 2:

Customer	Someone who buys products from Clear View Training Limited.
Shopkeeper	A user of the system who is responsible for managing the catalog of products.
SystemAdministrator	A special user of the system who can set up access rights for other users.
Dispatcher	A worker in the dispatch department of Clear View Training.
Inventory	Clear View Training's inventory system.
CardProcessingCompany	An external company that processes credit card transactions on behalf of Clear View Training.

3.4 Use Case Brief Semantics

The table below contains brief semantics for the use cases in the ECP system.

Table 3:

Use case	Brief semantics
ManageCustomerAccount	The Customer actor may create, update or destroy customer details held by the system. These details include name and address and user name and password.
ManageOrders	The Customer actor may create, view and change orders. An order may only be changed if it is still open.
ManageBasket	The Customer actor may add and remove items from their shopping basket.
Checkout	The Customer actor may check out. This involves paying for the items in the basket, and raising an order for the items.
BrowseProducts	The Customer actor may browse the products on sale.
FindProducts	The Customer actor may find one or more products based on some search criteria.
LogOnCustomer	The Customer actor must log on to the system before checking out or viewing /changing customer details
LogOnUser	The Shopkeeper, SystemAdministrator and Dispatcher actors must log on to the system before they can use it.

Table 3:

ManageProducts	The Shopkeeper actor may add and remove products from the catalog. The Shopkeeper actor may also update product details including the price.
ManageUsers	The SystemAdministrator actor may add, update or delete accounts for users of the system who are not Customers.
CloseOrder	The Dispatcher actor may set the status of an order to “closed” once the order has been shipped.

4 ECP Project Glossary version 1.1

Term	Definition
Catalog	A listing of all of the products that Clear View Training currently offers for sale.
Checkout	An electronic analogue of a real-world checkout in a supermarket. A place where customers can pay for the products in their shopping basket.
Clear View Training	A limited company specialising in sales of books and CDs.
Credit card	A card such as VISA or Mastercard that can be used for paying for products. Synonyms: Card
Customer	A party who buys products or services from Clear View Training Limited.
Dispatch Department	The department that is responsible for processing orders by assembling the required product items and shipping them to the customer.
Dispatch System	The system that the Dispatch Department uses to handle order processing. It integrates with the Inventory system.
Inventory System	The computer system that Clear View Training uses to manage its stock of products.
Order	A document raised by a customer specifying one or more products that have been purchased. The order specifies the quantity of each product, tax and shipping charges. Orders are passed to the Dispatch Department for processing. Orders are kept for five years to provide an audit trail of Clear View Training's sales activities.
Product	An item offered for sale by Clear View Training.
Product category	Products are assigned to categories to allow similar products (such as science fiction books) to be grouped together and to increase the ease of searching and maintenance of the catalogue.
Shopping basket	An electronic analogue of a real-world shopping basket. A place where customers can store their items prior to purchase.
Web browser	A program such as Internet Explorer 6 or Netscape Navigator which allows users to browse the World Wide Web.
User	Someone who uses the system but who is *not* a customer.

5 Use Case Model (Detailed)

5.1 Introduction

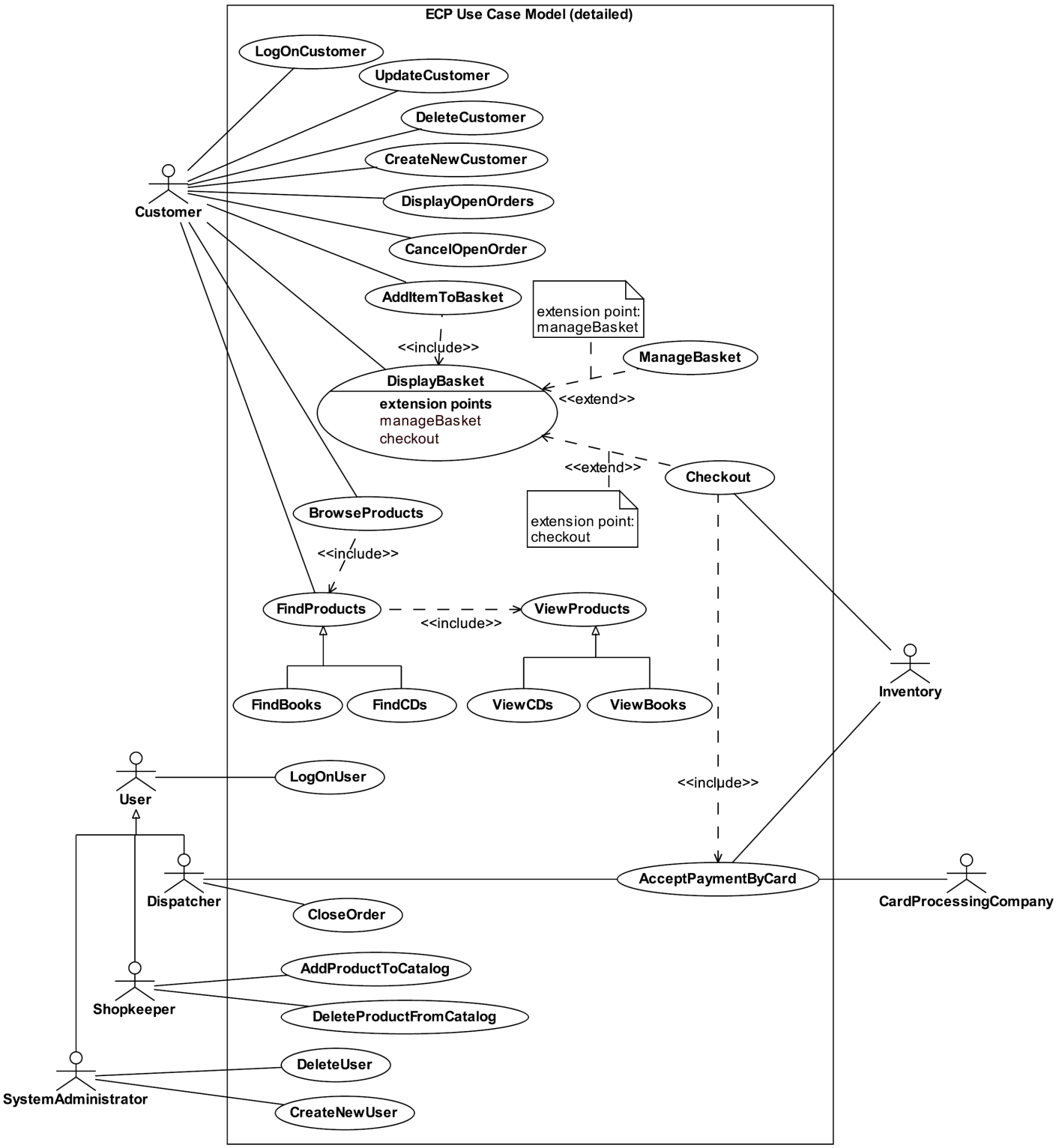
This document contains the detailed use case model for the ECP system. The model includes:

- Detailed use case diagram
- Actor semantics
- Detailed semantics for all use cases

5.2 Use case diagram (detailed)

The final use case diagram for the ECP system is shown in Figure 2:

Figure 2:


ECP Use Case Model (detailed)
LogOnCustomer
UpdateCustomer
DeleteCustomer
CreateNewCustomer
DisplayOpenOrders
CancelOpenOrder
AddItemToBasket
extension point:
manageBasket
<<include>>
ManageBasket
DisplayBasket
extension points
manageBasket
checkout
<<extend>>
<<extend>>
Checkout
BrowseProducts
extension point:
checkout
<<include>>
FindProducts
<<include>>
ViewProducts
FindBooks
FindCDs
ViewCDs
ViewBooks
Inventory
Customer
User
LogOnUser
<<include>>
Dispatcher
AcceptPaymentByCard
CardProcessingCompany
CloseOrder
AddProductToCatalog
Shopkeeper
DeleteProductFromCatalog
DeleteUser
SystemAdministrator
CreateNewUser

5.3 Actor semantics

The table below contains brief semantics for the actors in the ECP system.

Table 4:

Actor	Semantics
Customer	Someone who buys products from Clear View Training Limited.
Shopkeeper	A User of the system who is responsible for managing the catalogue of products.
SystemAdministrator	A special User of the system who can set up access rights for other Users.
Dispatcher	A User of the system that is worker in the dispatch department of Clear View Training.
User	Someone who uses the system but who is not a Customer.
Inventory	Clear View Training's inventory system.
CardProcessingCompany	An external company that processes credit card transactions on behalf of Clear View Training.

5.4 Use case detailed semantics

5.4.1 AcceptPaymentByCard use case

Extension Use Case: AcceptPaymentByCard
Use case ID: 1
Actors: Customer CardProcessingCompany Inventory Dispatcher
Preconditions: 1 The Customer is logged on to the system. 2 Inventory items have been provisionally reserved for the customer.
Primary scenario: 1 The use case begins when the Customer accepts the order. 2 The system retrieves the Customer's credit card details. 3 The system sends a message to the CardProcessingCompany that includes: merchant identifier, merchant authentication, name on card, number of card, expiry date of card, amount of transaction. 4 The CardProcessingCompany authorises the transaction. 5 The system notifies the Customer that the card transaction has been accepted. 6 The system gives the Customer an order reference number for tracking the order. 7 The system tells the Inventory actor to reserve the required items. 8 The system sends the order to the Dispatcher. 9 The system changes the order's state to pending. 10 The system displays an order confirmation that the Customer may print out.
Secondary scenarios: CreditLimitExceeded BadCard CreditCardPaymentSystemDown
Postconditions: 1 The order status has been set to pending. 2 The Customer's credit card has been debited by the appropriate amount. 3 Inventory items have been reserved to cover the order. 4 The order has been sent to the Dispatcher.

Secondary scenario: CreditLimitExceeded

1 The secondary scenario begins after step 3 of the primary scenario
2 The CardProcessingCompany does not authorise the transaction because the credit limit on the card would be exceeded.
3 The system displays a message telling the Customer that there was not enough credit available on the card to pay for the order.
4 The system displays a message asking the Customer to change their card details or contact their credit card company to extend their credit rating.
5 The system tells the Inventory actor to release the inventory reserved for this order.

Postconditions:

1 Inventory that had been provisionally reserved for this order is released.

Secondary scenario: BadCard

1 The secondary scenario begins after step 3 of the primary scenario
2 The CardProcessingCompany does not authorise the transaction because the credit card is invalid.
3 The system displays a message telling the Customer that their credit card is invalid.
4 The system displays a message asking the Customer to change their card details or contact their credit card company to get a new card.
5 The system tells the Inventory actor to release the inventory reserved for this order.

Postconditions:

1 Inventory that had been provisionally reserved for this order is released.

Secondary scenario: CreditCardPaymentSystemDown

1 The secondary scenario begins after step 3 of the primary scenario
2 The CardProcessingCompany can’t be contacted.
3 The system displays a message telling the Customer that their order can’t be processed at this point in time and to try again later.
4 The system tells the Inventory actor to release the inventory reserved for this order.

Postconditions:

1 Inventory that had been provisionally reserved for this order is released.

5.4.2 AddItemToBasket use case

Use case: AddItemToBasket
Use case ID: 2
Actors: Customer
Preconditions: 1 The Customer is browsing products.
Flow of events: 1 The Customer selects a product 2 The Customer selects “Add Item”. 3 The system adds the item to the Customer’s shopping basket. 4 include(DisplayBasket)
Postconditions: 1 A product has been added to the Customer’s basket. 2 The contents of the basket are displayed.

5.4.3 AddProductToCatalog use case

Use case: AddProductToCatalog
Use case ID: 3
Actors: Shopkeeper
Preconditions: 1 The Shopkeeper is logged on to the system.
Flow of events: 1 The use case begins when the Shopkeeper selects “Add Product”. 2 The system asks the Shopkeeper whether the product is a book or a CD. 3 If the product is a book 3.1 The system asks the Shopkeeper to enter the following product information: ISBN, title, category, authors, publisher, price, description, image URI. 4 If the product is a CD 4.1 The system asks the Shopkeeper to enter the following product information: title, category, artist, composer, label and catalog number, price, description, image URI. 5 The Shopkeeper enters the requested information. 6 The system asks the Shopkeeper to confirm that the new product should be added to the catalog. 7 The Shopkeeper confirms that the new product should be added. 8 The system adds the new product to the catalog.
Postconditions: 1 A new product has been added to the catalog.

5.4.4 BrowseProducts use case

Abstract use case: BrowseProducts
Use case ID: 4
Actors: Customer
Preconditions: None
None
Flow of events: 1 The use case begins when the Customer selects a type of product to browse (book or CD). 2 The system displays a list of all categories for that type of product. 3 The Customer selects a category. 4 include(ViewProducts)
Postconditions: None

5.4.5 CancelOpenOrder use case

Use case: CancelOpenOrder
Use case ID: 5
Actors: Customer Inventory
Preconditions: 1 The Customer is logged on to the system. 2 The system is displaying full details of an open order.
Flow of events: 1 The use case begins when the Customer selects “Cancel Order”. 2 The system asks the Customer for confirmation. 3 The Customer confirms the cancellation. 4 The system marks the order as cancelled. 5 The system records the time and date that the order was cancelled. 6 The system tells the inventory actor to release the inventory items reserved for this order.
Postconditions: 1 The order is cancelled. 2 Inventory items reserved for the order are released.

5.4.6 CloseOrder use case

Use case: CloseOrder
Use case ID: 6
Actors: Dispatcher
Preconditions: 1 The order has been forwarded to the Dispatcher. 2 The Dispatcher is logged on to the system. 3 The order status is “pending”. 4 The order has been processed by the Dispatch Department.
Flow of events: 1 The use case begins when the Dispatcher selects “Close Order”. 2 The system asks the Dispatcher to enter an order number, the date on which the order was shipped, the shipping company and tracking number if available. 3 The Dispatcher enters the requested information. 4 The system finds the order and changes the status of the order to “closed”. 5 The system adds to the order the date on which the order was shipped, the shipping company and tracking number (if available).
Postconditions: 1 The order status has been changed to “closed”. 2 The order has been updated to include the date on which the order was shipped, the shipping company and tracking number (if available).

5.4.7 Checkout use case

Extension Use Case: Checkout
Use case ID:8
Actors: Customer Inventory
Preconditions: 1 The Customer is logged on to the system.
Postconditions: 1 The use case begins when the Customer selects “Checkout”. 2 The system asks the Inventory actor to provisionally reserve the items in the shopping basket. 3 For each item that is out of stock 3.1 The system informs the Customer that the item is currently unavailable and it is removed from the order. 4 The system presents the final order to the Customer. The order includes an order line for each product that shows the product identifier, the product name, the quantity, the unit price, the total price for that quantity. The order also includes the shipping address and credit card details of the Customer and the total cost of the order including tax and postage and packing. 5 The system asks the Customer to accept or decline the order 6 The Customer accepts the order. 7 include(AcceptPaymentByCard)
Postconditions: 1 The Customer has accepted the order. 2 The ordered items have been reserved by the Inventory actor.

5.4.8 CreateNewCustomer use case

Use case: CreateNewCustomer
Use case ID: 9
Actors: Customer
Preconditions: 1 The Customer is not logged on to the system.
Flow of events 1 The use case begins when the Customer selects “New Customer”. 2 The system asks the Customer to enter a user name and password. 3 The Customer enters the requested information. 4 The system checks to see if the user name is available and the password is valid. 5 While the username is not available or the password is invalid 5.1 The system asks for a new user name and/or password. 6 The system asks the Customer for the following information: name and address (mandatory), email address (mandatory), phone number (optional), fax number (optional), credit card details (mandatory). 7 The Customer enters the requested information. 8 While mandatory information is missing 8.1 The system asks the Customer for the missing information. 8.2 The Customer enters the missing information. 9 The system confirms that the customer information has been accepted. 10 The system assigns a unique customer identifier to the customer.
Postconditions: 1 The system has saved the Customer details. 2 The Customer is assigned a user name and password. 3 The Customer is assigned a unique customer identifier.

5.4.9 CreateNewUser use case

Use case: CreateNewUser
Use case ID: 10
Actors: SystemAdministrator
Preconditions: 1 The SystemAdministrator is logged on to the system.
Flow of events: 1 The use case begins when the SystemAdministrator selects “New User”. 2 The system asks the SystemAdministrator to input the following information: user name, password, name of user, email address. 3 The system creates a new user account.
Postconditions: 1 A new user account has been created.

5.4.10 DeleteCustomer use case

Use case: DeleteCustomer
Use case ID: 11
Actors: Shopkeeper
Preconditions: 1 The Shopkeeper is logged on to the system.
Flow of events: 1 The Shopkeeper selects “Delete Customer”. 2 The system asks the Shopkeeper for a customer identifier. 3 The Shopkeeper enters the customer identifier. 4 The system displays the customer details. 5 The Shopkeeper confirms the deletion. 6 The system deletes the customer details.
Postconditions: 1 The customer details have been deleted from the system.

5.4.11 DeleteProductFromCatalog use case

Use case: DeleteProductFromCatalog
Use case ID: 12
Actors: Shopkeeper
Preconditions: 1 The Shopkeeper is logged on to the system.
Flow of events: 1 The use case begins when the Shopkeeper selects “Delete Product”. 2 The system asks the Shopkeeper for the product identifier of the product to delete. 3 The Shopkeeper enters the product identifier. 4 The system displays the product details. 5 The Shopkeeper cofirms the deletion. 6 The system deletes the product from the catalog.
Postconditions: 1 A product has been deleted from the catalog.

5.4.12 DeleteUser use case

Use case: DeleteUser
Use case ID: 13
Actors: SystemAdministrator
Preconditions: 1 The SystemAdministrator is logged on to the system.
Flow of events: 1 The use case begins when the SystemAdministrator selects “Delete User”. 2 The system asks the Shopkeeper for a user name. 3 The Shopkeeper enters the user name. 4 The system displays the user details. 5 The Shopkeeper confirms the deletion. 6 The system deletes the user’s account.
Postconditions: 1 A user’s account has been deleted.

5.4.13 DisplayBasket use case

Use case: DisplayBasket
Use case ID: UC14
Actors: Customer
Preconditions: None.
Flow of events: 1 The Customer selects “Display Basket”. 2 If there are no items in the basket 2.1 The system tells the Customer that the basket is empty. 2.2 The use case terminates 3 For each product in the basket 3.1 The system displays the product id, quantity, details, unit price and total price. extension point: manageBasket extension point: checkout
Postconditions: None.

5.4.14 DisplayOrders use case

Use case: DisplayOpenOrders
Use case ID: UC15
Actors: Customer
Preconditions: 1 The Customer is logged on to the system.
Flow of events: 1 The use case begins when the Customer selects “Display Open Orders”. 2 The system displays a one line summary of each open order the Customer has made. Orders are displayed in chronological order, and the information on each line is the order date, number, summary, amount and status (which is always open in this use case). 3 The Customer selects an order. 4 The system displays full details of the order.
Postconditions: 1 The system displays full details of an order.

5.4.15 FindBooks use case

Use case: FindBooks
Use case ID: UC16
Parents: FindProducts
Actors: Customer
Preconditions: None
Flow of events: *1 The Customer selects "Find Book".* *2 The system asks the Customer for book search criteria that consist of one or more of the following: title, author, ISBN, category.* 3 The Customer enters the search criteria. *4 The system searches for books that match the Customer's criteria.* *5 If the system finds some books* *5.1 include(ViewBooks).* 6 Else 6.1 The system tells the Customer that no matching products were found.
Postconditions: None

5.4.16 FindCDs use case

Use case: FindCDs
Use case ID: UC17
Parents: FindProducts
Actors: Customer
Preconditions: None
Flow of events: *1 The Customer selects "Find CD".* *2 The system asks the Customer for CD search criteria that consists of one or more of the following: title, artist, composer, label and catalog number, category* 3 The Customer enters the search criteria. *4 The system searches for CDs that match the Customer's criteria.* *5 If the system finds some products* *5.1 include(ViewCDs).* 6 Else 6.1 The system tells the Customer that no matching products were found.
Postconditions: None

5.4.17 FindProducts use case

Abstract use case: FindProducts
Use case ID: UC18
Actors: Customer
Preconditions: None
Flow of events: 1 The Customer selects “Find”. 2 The system asks the Customer for search criteria. 3 The Customer enters the search criteria. 4 The system searches for products that match the Customer's criteria. 5 If the system finds some products 5.1 include(ViewProducts) 6 Else 6.1 The system tells the Customer that no matching products were found.
Postconditions: None

5.4.18 LogOnCustomer use case

Use case: LogOnCustomer
Use case ID: UC19
Actors: Customer
Preconditions: 1 The Customer is not logged on to the system.
Primary scenario: 1 The use case begins when the Customer actor accesses the first page of the website. 2 The system automatically recognises the Customer. 3 The system automatically logs the Customer on to the site.
Secondary scenarios: ExistingCustomerNotRecognised CustomerAuthenticationFails
Postconditions: 1 The Customer is logged on to the system.
Secondary scenario: ExistingCustomerNotRecognised
Preconditions: 1 The Customer has not been recognised by the system.
Secondary scenario: 1 The scenario begins when the Customer selects “Log On”. 2 While the Customer is not logged on and the number of authentication attempts is less than or equal to three 2.1 The system asks the Customer for their user name and password. 2.2 The Customer enters their user name and password. 2.3 The user name and password are correct. 3 The system authenticates the Customer.
Postconditions: 1 The Customer is logged on to the system.
Secondary scenario: CustomerLogOnFails
Preconditions: 1 The Customer has not been recognised by the system.

Secondary scenario:

1 The scenario begins when the Customer selects “Log On”.
2 While the Customer is not logged on and the number of authentication attempts is less than or equal to three
    2.1 The system asks the Customer for their user name and password.
    2.2 The Customer enters their user name and password.
    2.3 The user name and/or password are wrong.
3 The system logs a security violation.

Postconditions:

1 The Customer is not authenticated.
2 The system has logged a security violation.

5.4.19 ManageBasket use case

Extension Use Case: ManageBasket
Use case ID: UC21
Actors: Customer
Preconditions: 1 The System is displaying the shopping basket.
Flow of events: 1 While the Customer is updating the basket 1.1 The Customer selects an item in the basket. 1.2 If the Customer selects “Remove Item” 1.2.1 The system displays the message “Are you sure you want to remove the selected item from your basket?”. 1.2.2 The Customer confirms the removal. 1.2.3 The system removes the selected item from the basket. 1.3 If the Customer enters a new quantity for the selected item 1.3.1 The system updates the quantity for the selected item.
Postconditions: None.

5.4.20 LogonUser use case

Use case: LogOnUser
Use case ID: UC20
Actors: User
Preconditions: 1. The User is not logged on to the system.
Primary scenario: 1 The use case begins when the User selects “Log On”. 2 While the User is not authenticated and the number of authentication attempts is less than or equal to three 2.1 The system asks the User for their user name and password. 2.2 The User enters their user name and password. 2.3 The user name and password are correct. 3 The system authenticates the User.
Postconditions: 1 The User is logged on.
Secondary scenario: UserAuthenticationFails
1 The scenario begins when the User selects “Log On”. 2 While the User is not authenticated and the number of authentication attempts is less than or equal to three 2.1 The system asks the User for their user name and password. 2.2 The User enters their user name and password. 2.3 The user name and/or password is wrong. 3 The system logs a security violation. 4 The system freezes the User account.
Postconditions: 1 The User is not logged on. 2 The system has logged a security violation. 3 The User’s account has been frozen pending investigation of the security violation.

5.4.21 UpdaeCustomer use case

Use case: UpdateCustomer
Use case ID: UC24
Actors: Customer
Preconditions: 1 The Customer is logged on to the system.
Flow of events: 1 The use case begins when the Customer selects “Update Customer Details”. 2 The system displays the customer details including name and address (mandatory), email address (mandatory), phone number (optional), fax number (optional), credit card details (mandatory). The optional elements may be empty if the Customer had not previously entered them. 3 While (not finished changing) 3.1 The Customer selects a field and changes its value. 4 The system asks the Customer to confirm the changes. 5 The Customer selects OK.
Postconditions: 1 The Customer’s details have been updated.

5.4.22 ViewBooks use case

Use case: ViewBooks
Use case ID: UC4
Parents: ViewProducts
Actors: Customer
Preconditions: *1. A set of books has been identified for browsing*
Flow of events: *1 The system displays a page containing a maximum of 10 books. This page includes the following summary information for each book: title, author, publisher, price.* 2 While the Customer is browsing 2.1 If there are more products to display 2.1.1 The Customer may select “Next” to view the next page of products. 2.2 If the Customer is not on the first page of products 2.2.1 The Customer may select “Previous” to view the previous page of products. 2.3 If the Customer selects a product 2.3.1 *The system displays full details of the book including ISBN, title, category, authors, publisher, price, description, image URI*. 2.3.2 The Customer may select “Back” to continue browsing.
Postconditions: None

5.4.23 ViewCDs use case

Use case: ViewCDs
Use case ID: UC5
Parents: ViewProducts
Actors: Customer
Preconditions: *1. A set of CDs has been identified for browsing*
Flow of events
1 The system displays a page containing a maximum of 10 CDs. This page includes the following summary information for each CD: title, artist, composer, price. 2 While the Customer is browsing 2.1 If there are more products to display 2.1.1 The Customer may select "Next" to view the next page of products. 2.2 If the Customer is not on the first page of products 2.2.1 The Customer may select "Previous" to view the previous page of products. 2.3 If the Customer selects a product 2.3.1 *The system displays full details of the CD including title, artist, composer, label and catalog number, category, image URI.* 2.3.2 The Customer may select "Back" to continue browsing.
Postconditions: None

Abstract use case: ViewProducts

Use case ID: UC6

Actors:

Customer

Preconditions:

1. A set of products has been identified for viewing

Flow of events:

1 The system displays a page containing the first 10 products. This page includes summary details for each product including its price.
2 While the Customer is browsing
- 2.1 If there are more products to display
  - 2.1.1 The Customer may select “Next” to view the next page of products.
- 2.2 If the Customer is not on the first page of products
  - 2.2.1 The Customer may select “Previous” to view the previous page of products.
- 2.3 If the Customer selects a product
  - 2.3.1 The system displays full details of the product including its price.
  - 2.3.2 The Customer may select “Back” to continue browsing

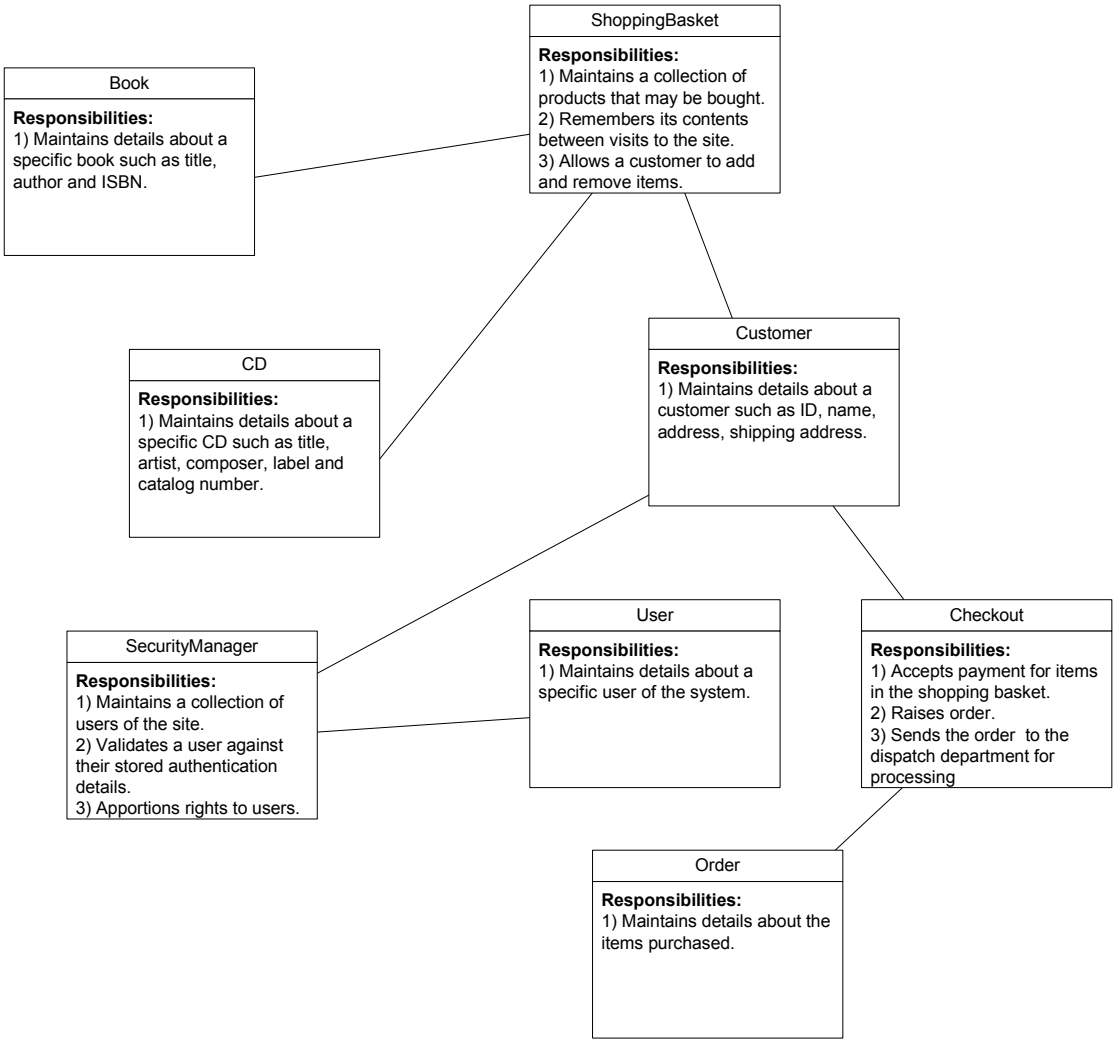
Postconditions:

None

5.5 CRC brainstorm

Figure 3 shows the result of the initial CRC brainstorm for the ECP system. Rather than using the more traditional class, responsibilities, collaborators format, we have only listed the class and the responsibilities in each card. The collaborations are represented by lines joining the cards together. This is our favourite approach, but it only works if you are using sticky notes on a white board or flip chart. Notice how the CRC model begins to look like a class diagram.

Figure 3:



5.6 Noun/verb analysis

Here is an example noun/verb analysis of the ECP Informal Specification. Noun/verb analysis should also be done on the use cases, and any other documents related to the ECP.

In this example, we have analysed the text looking for candidate classes, attributes and operations, and have marked up the text as follows:

Table 5:

Candidate	Markup
class	**bold**
attribute	underline
operation	*italic*

The advantage of this approach is that each of these types of markup can be combined. So, if, for example, you find a noun phrase that suggests a candidate class in the middle of a verb phrase that is a candidate operation, you may mark the verb phrase as italic, and the candidate class as bold. In the document, the candidate class will then be bold, italic.

If the document you are marking up already uses bold, italic or underline extensively, then you can choose a different style of markup, such as a different color, or even just use highlighers.

Clear View Training E Commerce Platform
Informal System Specification

Vision

The E-Commerce Platform (ECP) is a new web-based **selling channel** for Clear View Training Limited.

The goal of the ECP is to allow Clear View Training **customers** to order **products** via the Internet from an online **catalog**.

The ECP must integrate with the existing **inventory** and **dispatch** systems and must also *communicate* **credit card information** to the **credit card processing company** for *validation* before an **order** is *accepted*.

We believe that the system should operate according to the "**shopping basket**" paradigm that other very successful web stores such as Amazon.com use. In this model a **catalog** of **products** is *displayed* and the users can click on "*Add to basket*" to place a **product** in their **shopping basket**. This idea is demonstrated in the user interface prototype.

User Interface Prototype

The prototype is currently just a set of browser screens created in Microsoft Visio that can be found in the document "ECP User Interface Prototype.doc".

Clear View Training Products

At this time, **Clear View Training** only intends to *sell* **books** and **CDs** via its **ECP web channel**.

Books

Books are *categorized* according to subject matter. These categories include, but are not limited to:

Table 1 - Book Categories

Art	History	Reference
Biographies	Hobbies	Religion & spirituality
Children's books	Home & garden	Science & nature
Finance	Horror	Science fiction
Computers	Literature & fiction	Sports & outdoors
Cooking, food & wine	Mystery & thrillers	Travel
Entertainment	Non-fiction	
Mind & body	Professional & technical	

Each **book** is identified by its ISBN.

Customers can browse the **book catalog** by category or *find a given book* based on the following **search criteria**:

- Author
- Title
- Publisher
- ISBN

CDs

CDs are *categorized* according to subject matter. These categories include, but are not limited to:

Table 2 - CD categories

Alternative	International	Soul
Blues	Jazz	Soundtracks
Children's music	Miscellaneous	Vocalists & Broadway
Classical	New Age	World
Country	Opera & vocal	
Dance & DJ	Pop	
Folk	Rap & hip-hop	
Emerging artists	R&B	

Customers can browse the **CD catalog** by category or *find a given CD* based on the following **search criteria**:

- Artist
- Title
- Label
- Composer

Product Catalog

As the user interface prototype shows, we expect the ECP to offer the **customer** an initial choice of **book** or **CD**.

On *selecting either **book** or **CD*** the **ECP** should then list the categories and allow the customer to *choose a category* or *search for a specific **product***.

The result of *choosing a category* or *doing a **search*** is the same – a summary list of **products**:

• For **books** this **summary** should contain at least author, title, publisher, ISBN, price.
• For **CDs** this **summary** should contain at least artist, title, label, composer, price.

*Clicking on any **product** in the **summary*** will bring up a **full product description** that includes all of the **product information**, the price and an **optional picture**. Next to the price there is an "*Add to basket*" **button**.

The shopping basket

When an *item is added to the **shopping basket***, the **customer** is taken to the **shopping basket screen** that shows the **list of all products currently in the basket.** On this screen the **customer** may:

• *Remove an **item** from the **basket***
• *Change the quantity of an **item***
• *Proceed to **checkout***

Checkout

The **system** presents the **customer** with a **summary of their order**. If they click on "*confirm*" to *confirm the **order,*** then the **system** asks them to *log in* if they have not already done so.

Ideally, the **checkout** should recognize the **customer** in which case the *log in is automatic*.

If not, then existing **customers** must *log in* by *entering a user **name** and **password***.

New **customers** must fill out a **form** that asks for the following details:

• Name
• Address
• Shipping address (if different from above)
• Email address
• Phone number
• Fax number
• Credit card details

On s*ubmitting this form*, the **customer** will be issued with a **user name** (which should probably be their email address) and is asked to select a **password**.

Order processing then completes.

Here is a table containing the results of the analysis:

Table 6:

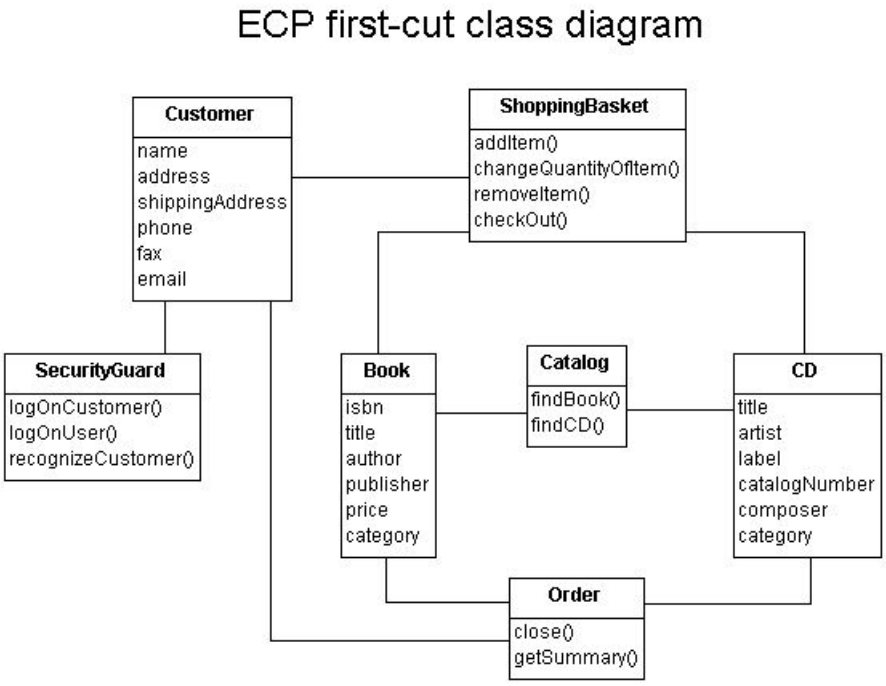
Candidate classes	Candidate attributes	Candidate operations
book catalog	Address	accepted
books	artist	Add to basket
catalog	Author	browse
CD catalog	catalog number	categorised
CDs	category	Change the quantity of an item

checkout	composer	choose a category
Clear View Training	Email address	Clicking on any product in the summary
Credit card details	Fax number	communicate
credit card information	ISBN	confirm
credit card processing company	label	displayed
customers	Name	doing a search
dispatch	Phone number	find a given book
ECP web channel	picture	find a given CD
form	price	item is added to the shopping basket
full product description	Publisher	list the categories
HTML pages	Shipping address	log in
inventory	subject matter	log in is automatic
list of all products currently in the basket	title	Order processing
order		Proceed to checkout
password		recognise the customer
product information		Remove an item from the basket
products		search for a specific product
search criteria		selecting either book or CD
selling channel		submitting this form
shopping basket		validation
shopping basket screen		
summary		
summary of their order		
system		
user name		
users		

5.7 Consolidation

Using the results of CRC brainstorming and noun/verb analysis, we can create a first-cut class diagram as shown in Figure 4.

Figure 4:



6 Use Case Realization

6.1 Introduction

The process of use case realization involves demonstrating how the analysis classes that you have identified interact together to realize the behavior specified by the use cases. Use case realizations consist of:

- Detailed analysis class diagrams
- Sequence and communication diagrams for the use cases

To perform use case realization you go through the key use cases and create interaction diagrams to illustrate their behavior. As you do this you identify:

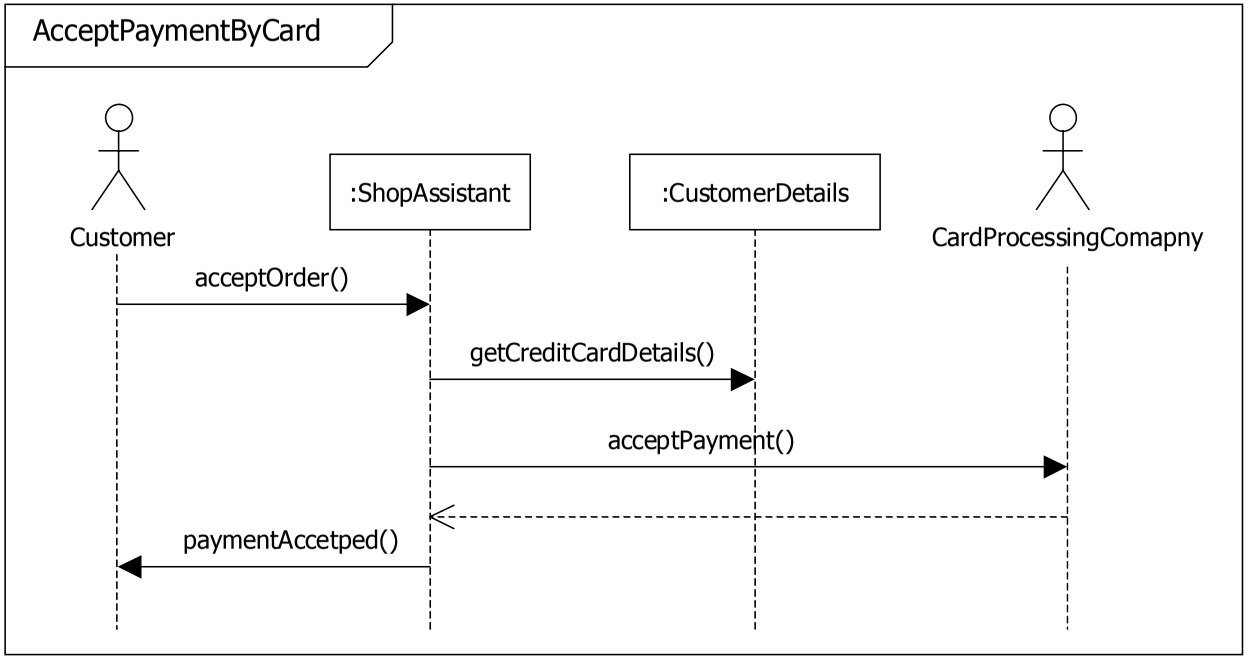
- Attributes and operations of existing analysis classes
- New analysis classes

This leads to a final analysis model comprising detailed class diagrams and a set of use case realizations.

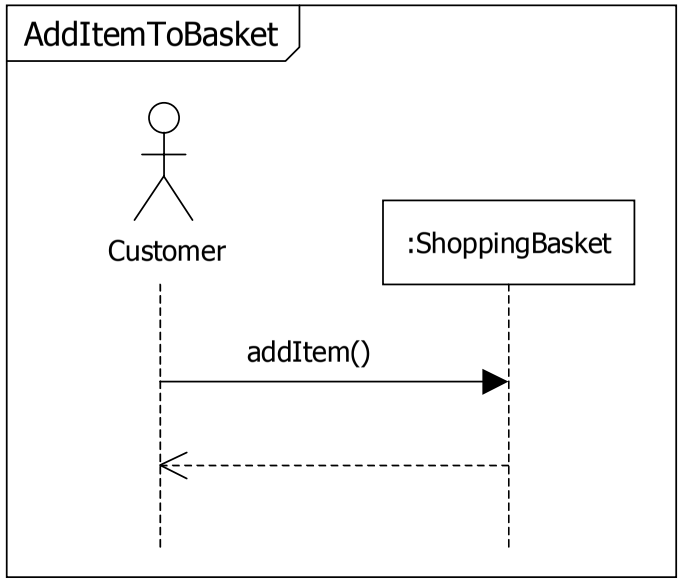
6.2 Interaction diagrams

Usually, you would only create interaction diagrams for the key use cases. For teaching purposes, we have created an interaction diagram for every use case in the model.

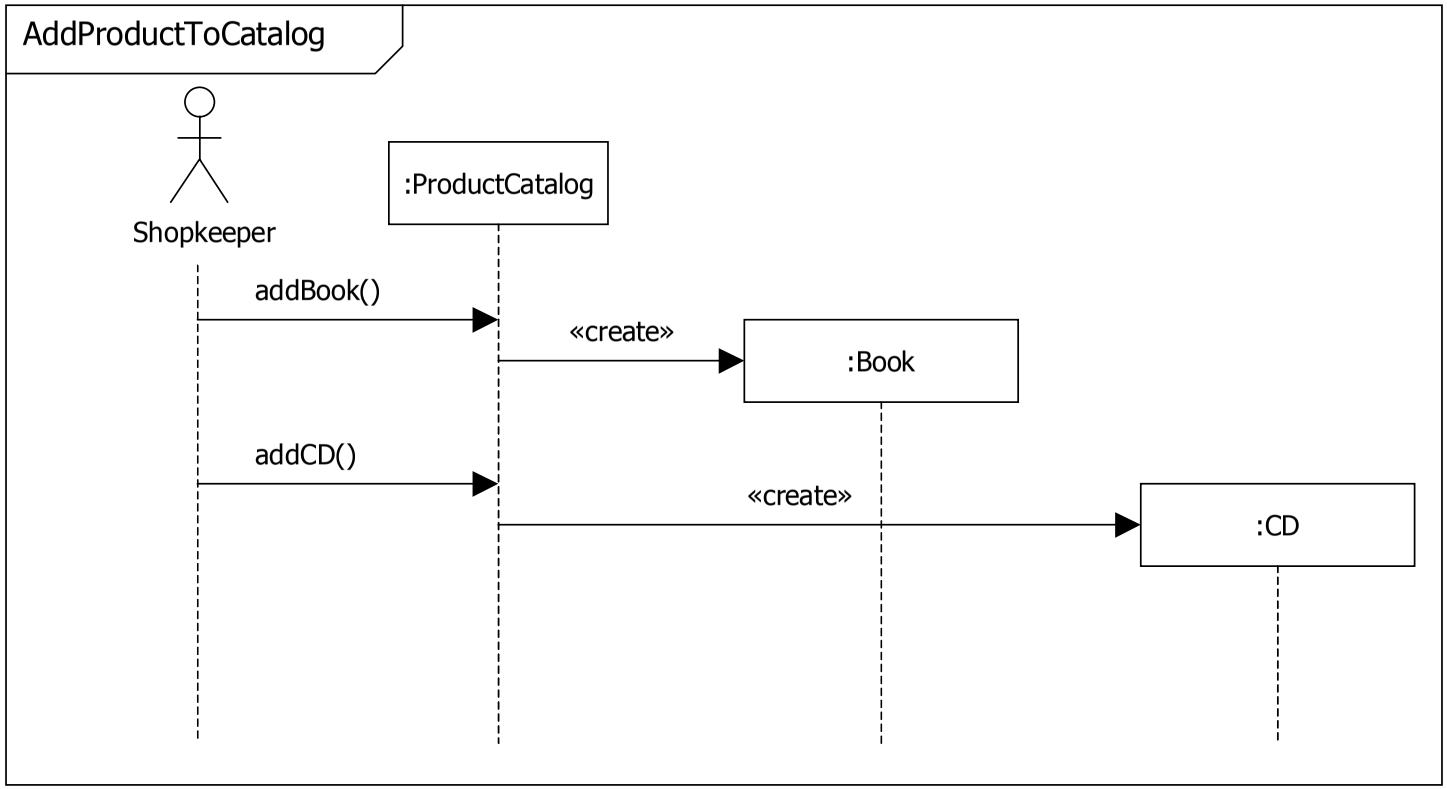
6.2.1 AcceptPaymentByCard Sequence Diagram



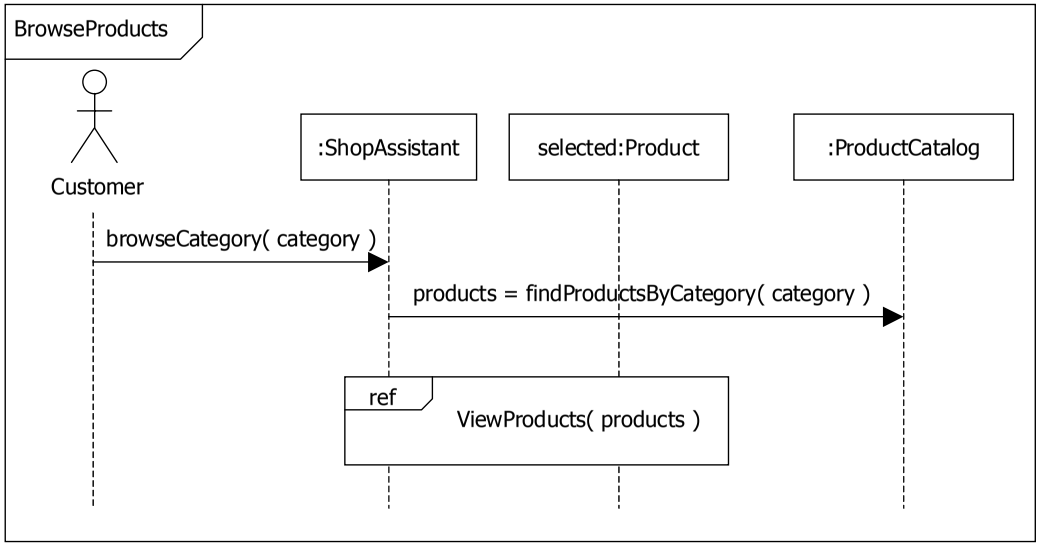
6.2.3 AddItemToBasket Sequence Diagram



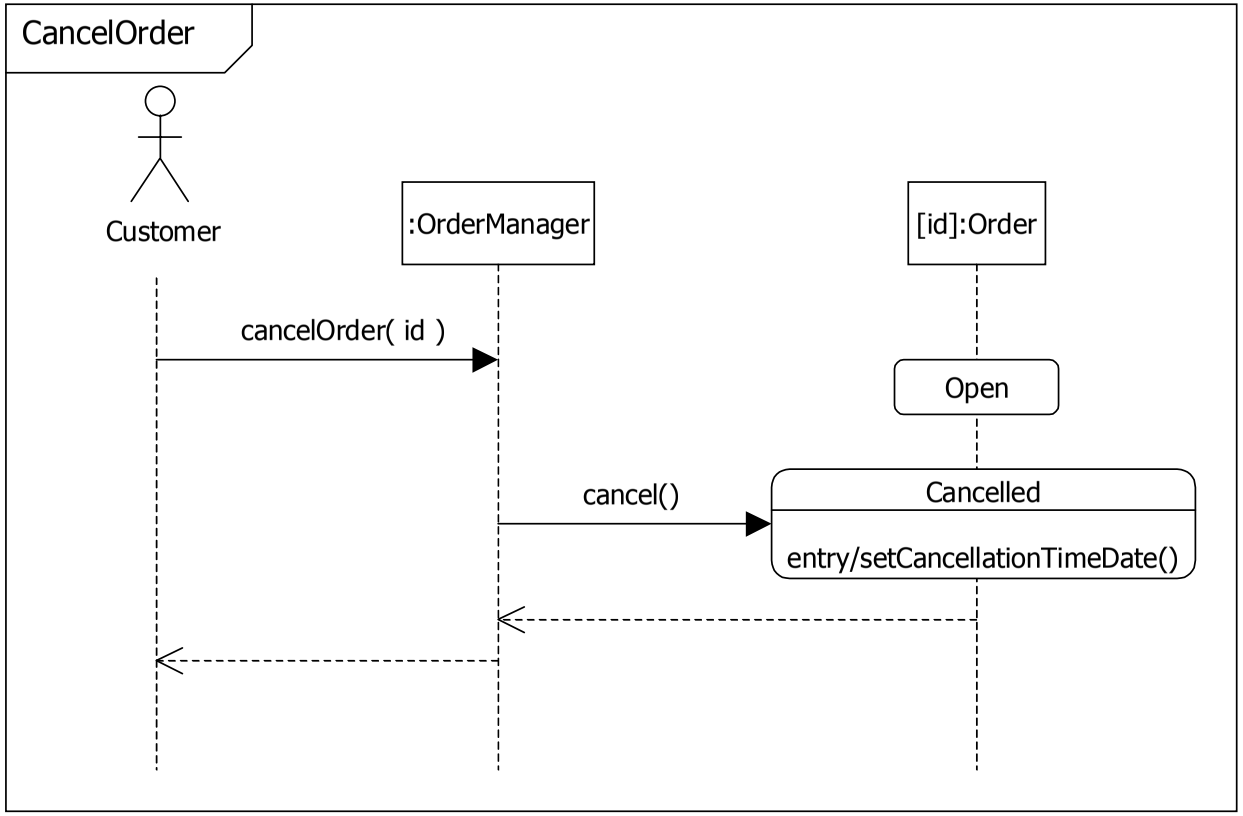
6.2.2 AddProductToCatalog Sequence Diagram



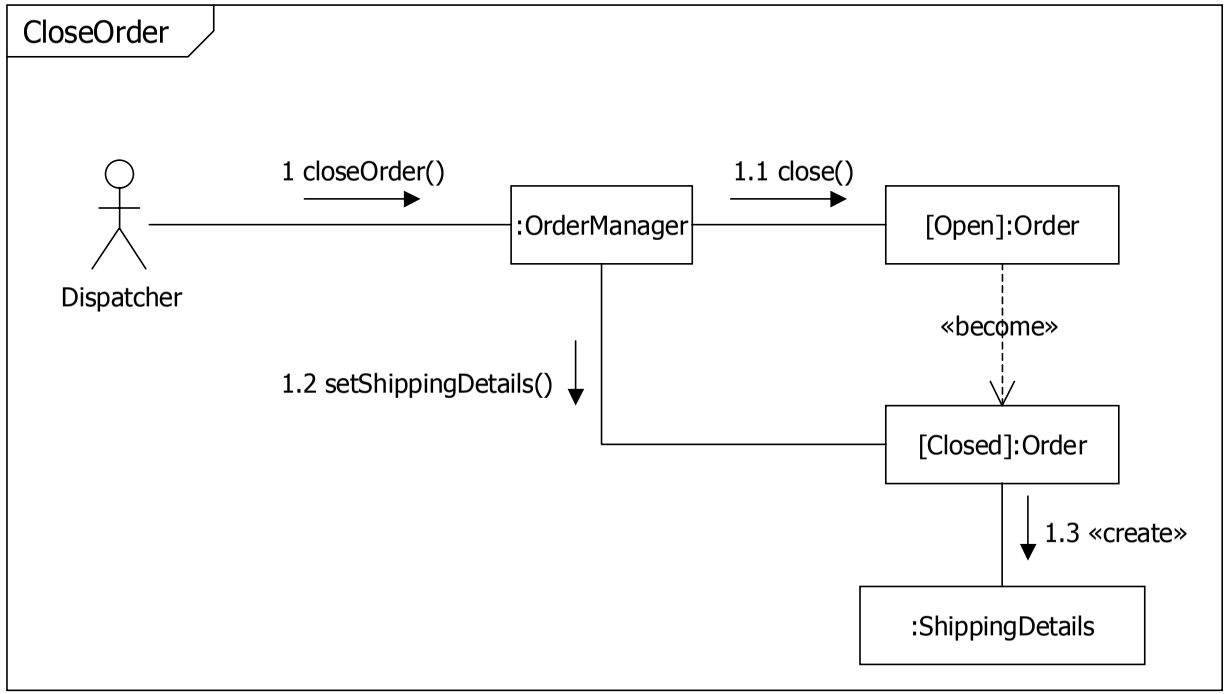
6.2.4 BrowseProducts Sequence Diagram



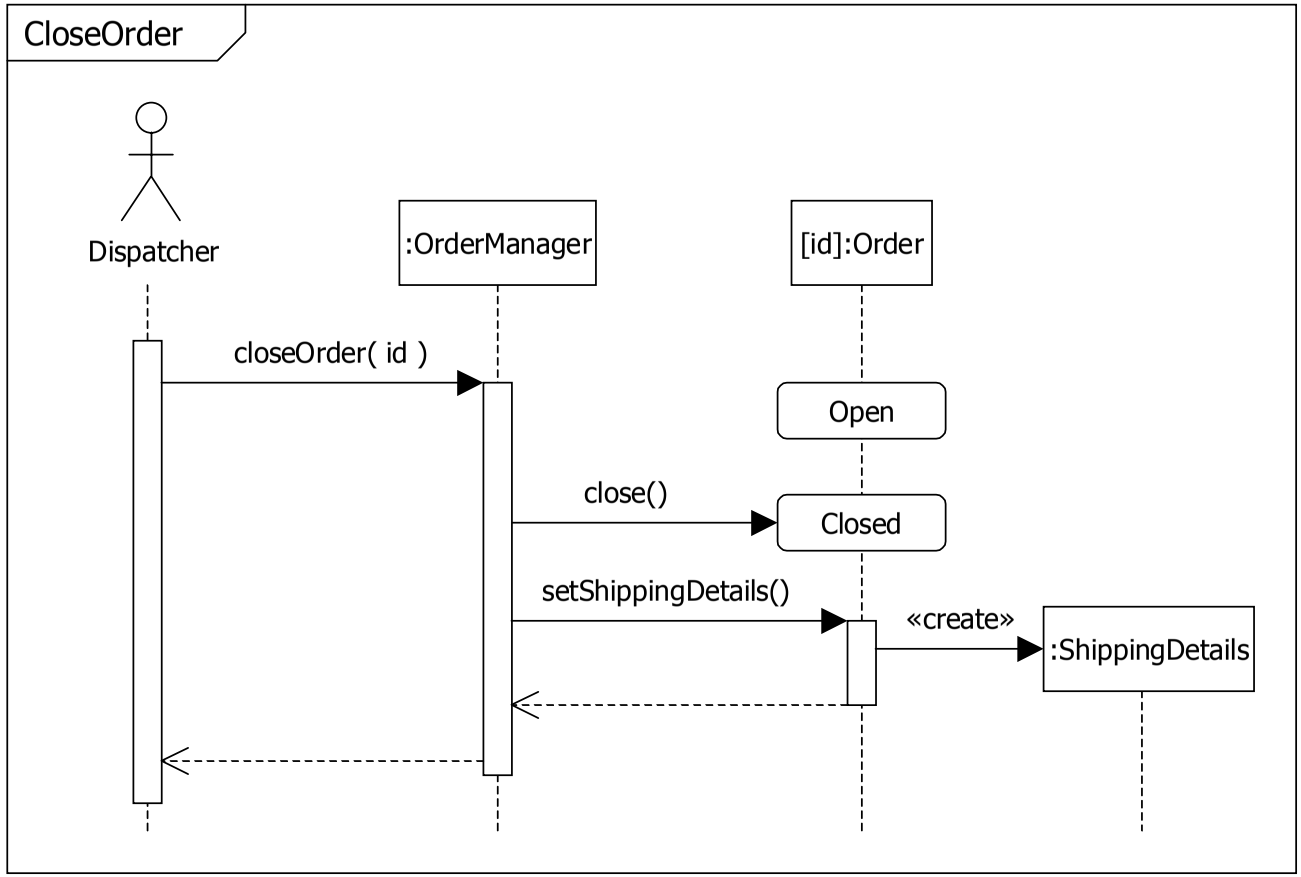
6.2.5 CancelOrder Sequence Diagram



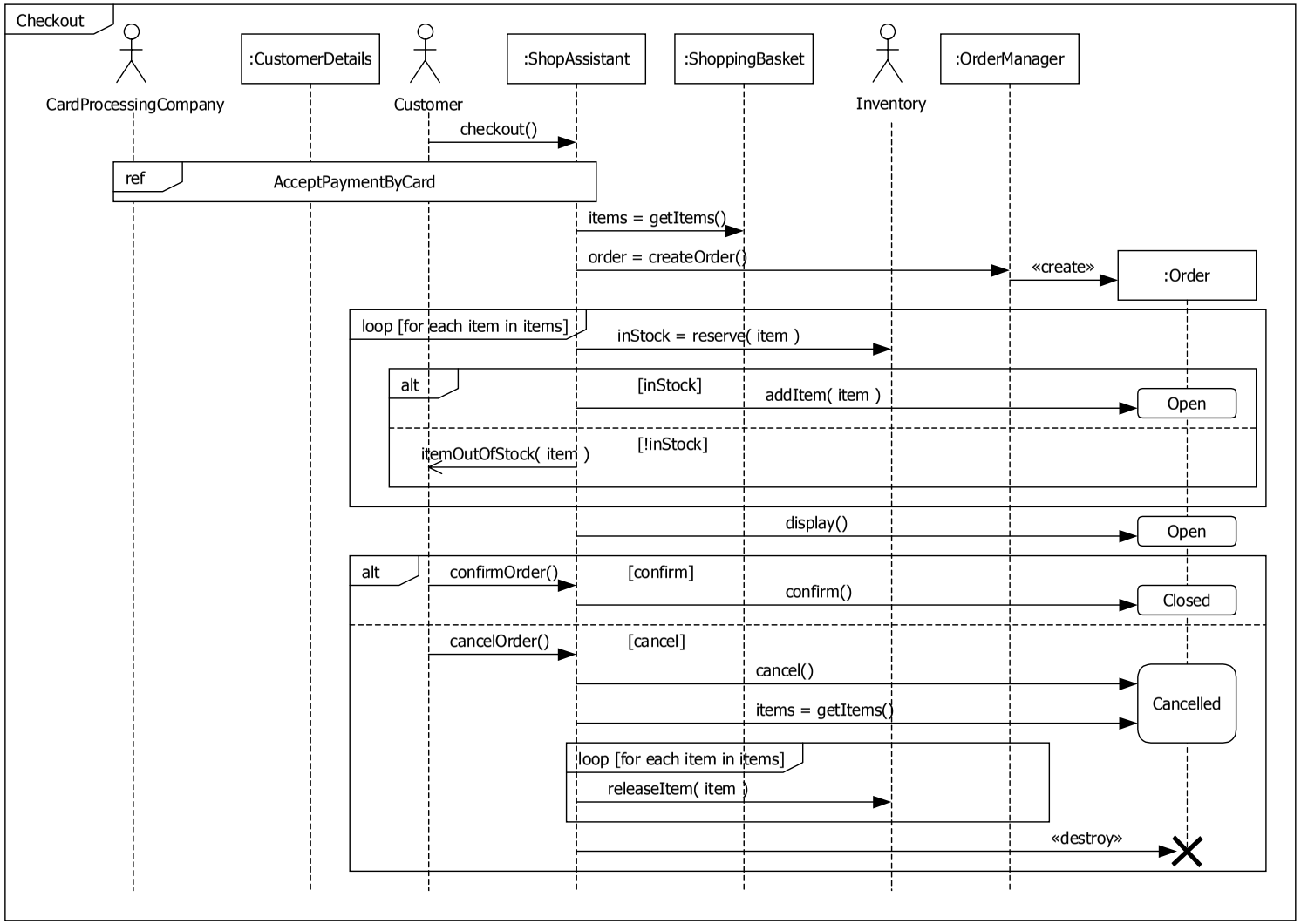
6.2.6 CloseOrder Communication Diagram



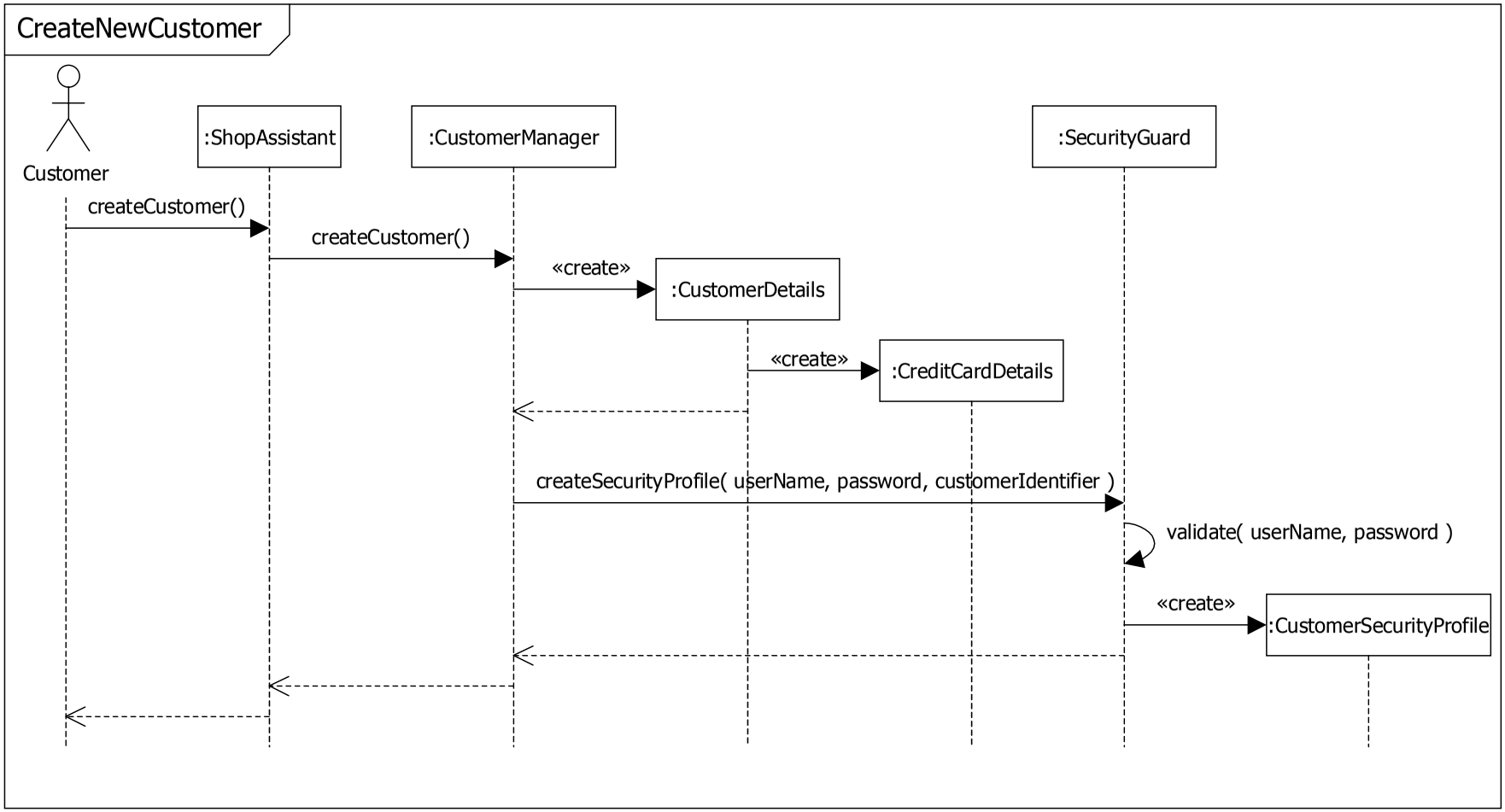
6.2.7 CloseOrder Sequence Diagram



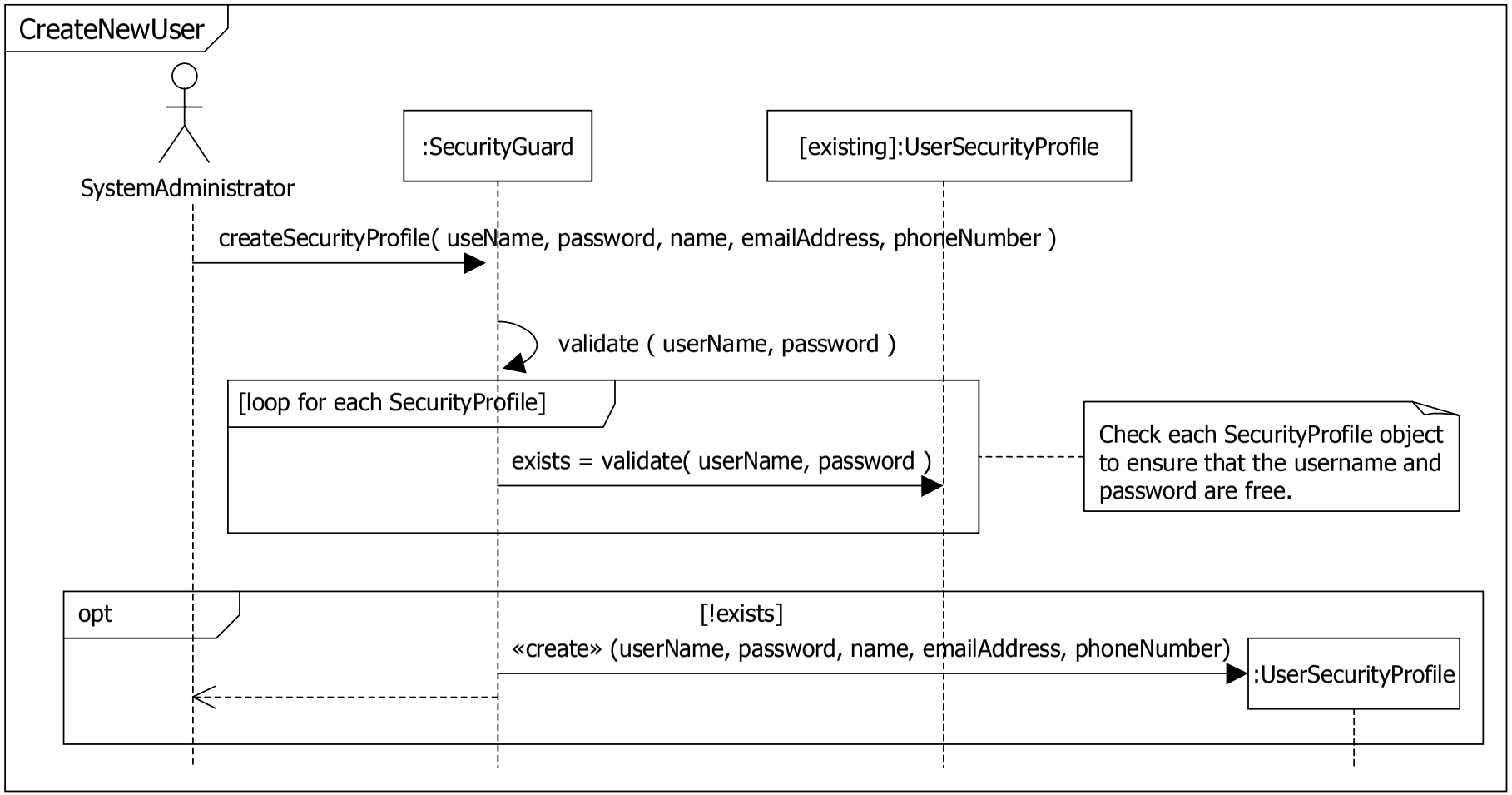
6.2.8 Checkout Sequence Diagram



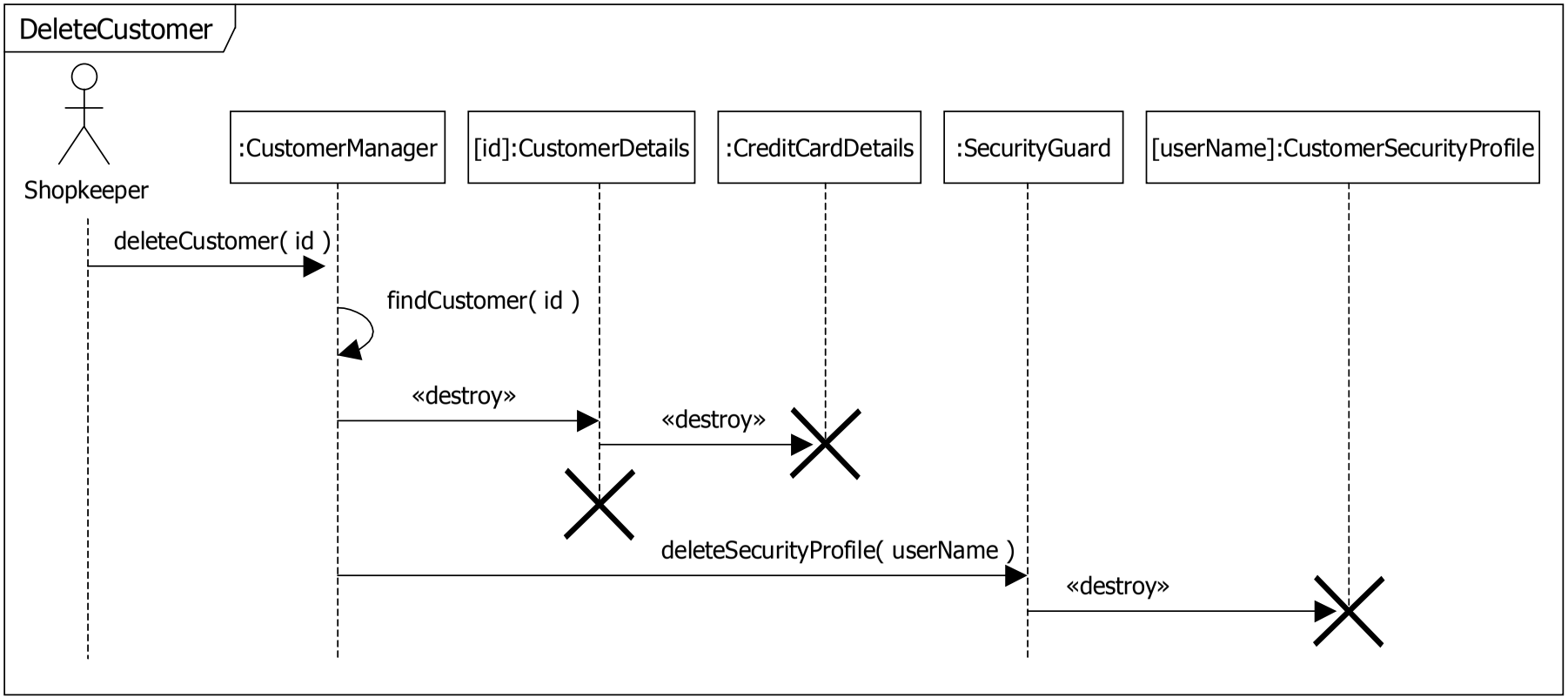
6.2.9 CreateNewCustomer Sequence Diagram



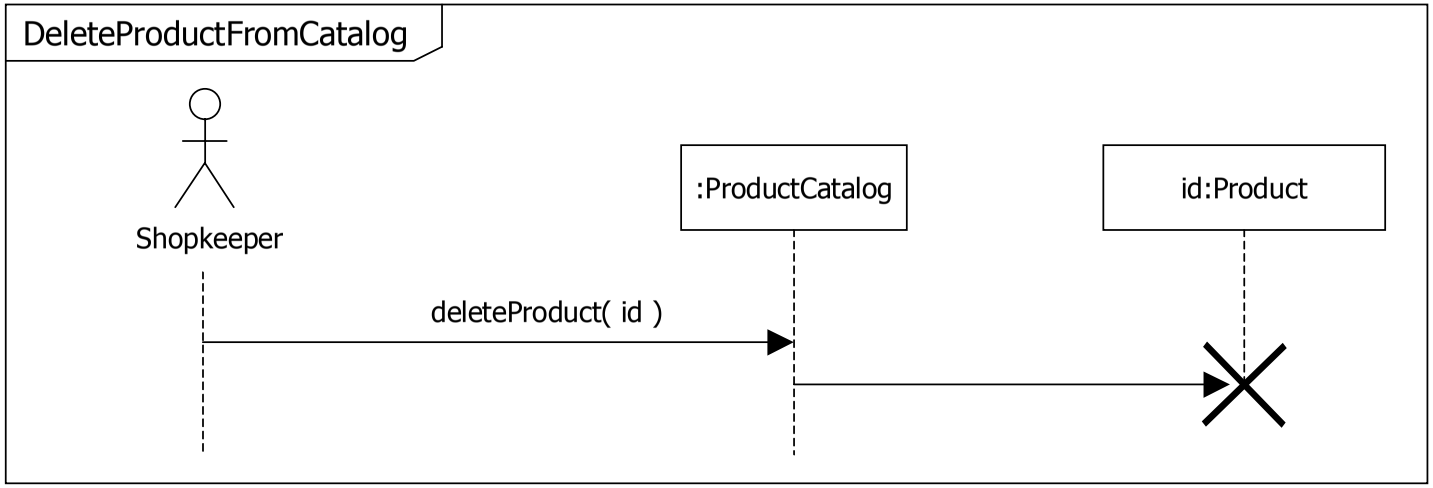
6.2.10 CreateNewUser Sequence Diagram



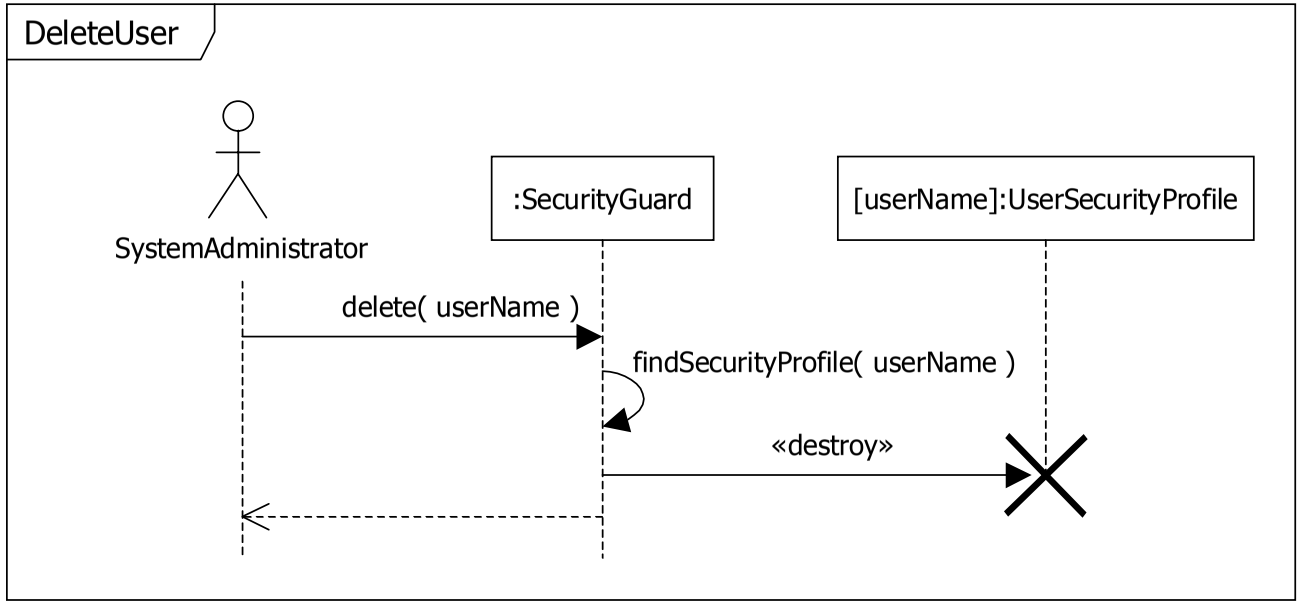
6.2.11 DeleteCustomer Sequence Diagram



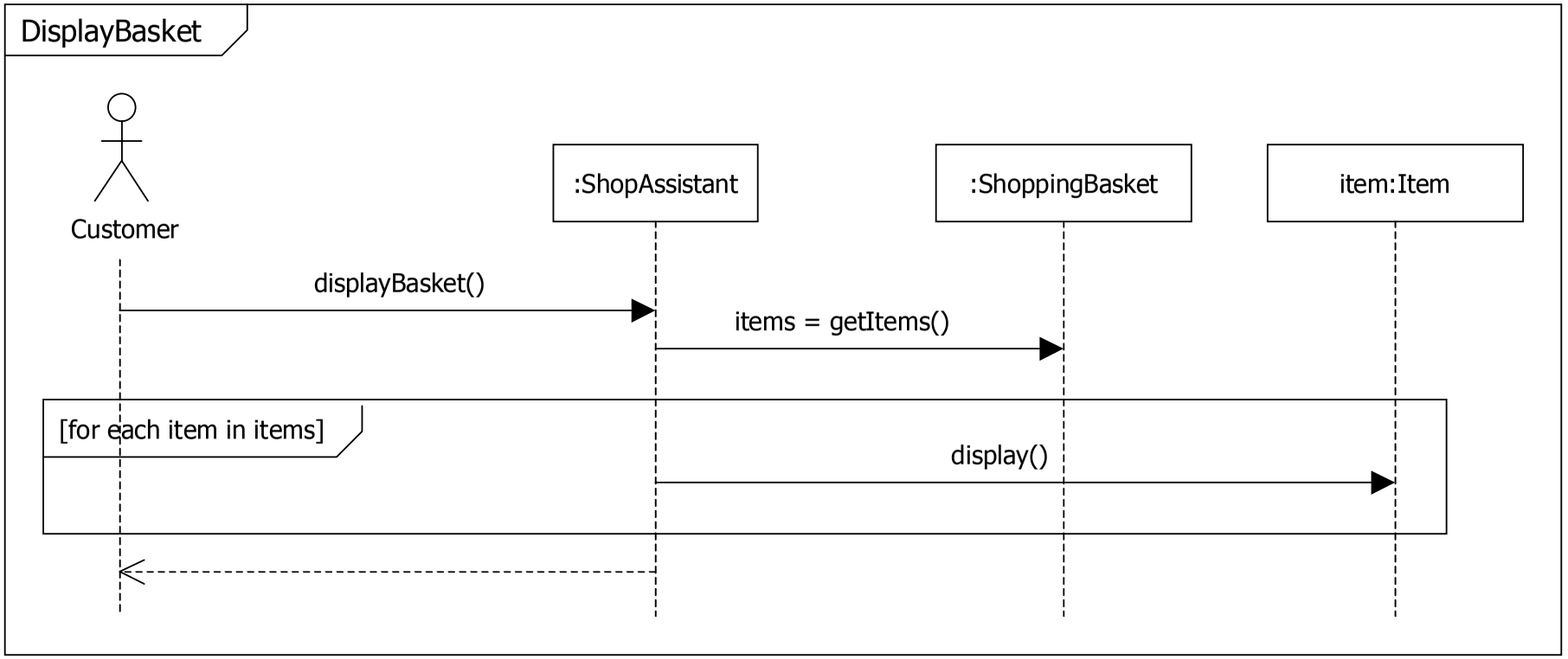
6.2.12 DeleteProductFromCatalog Sequence Diagram



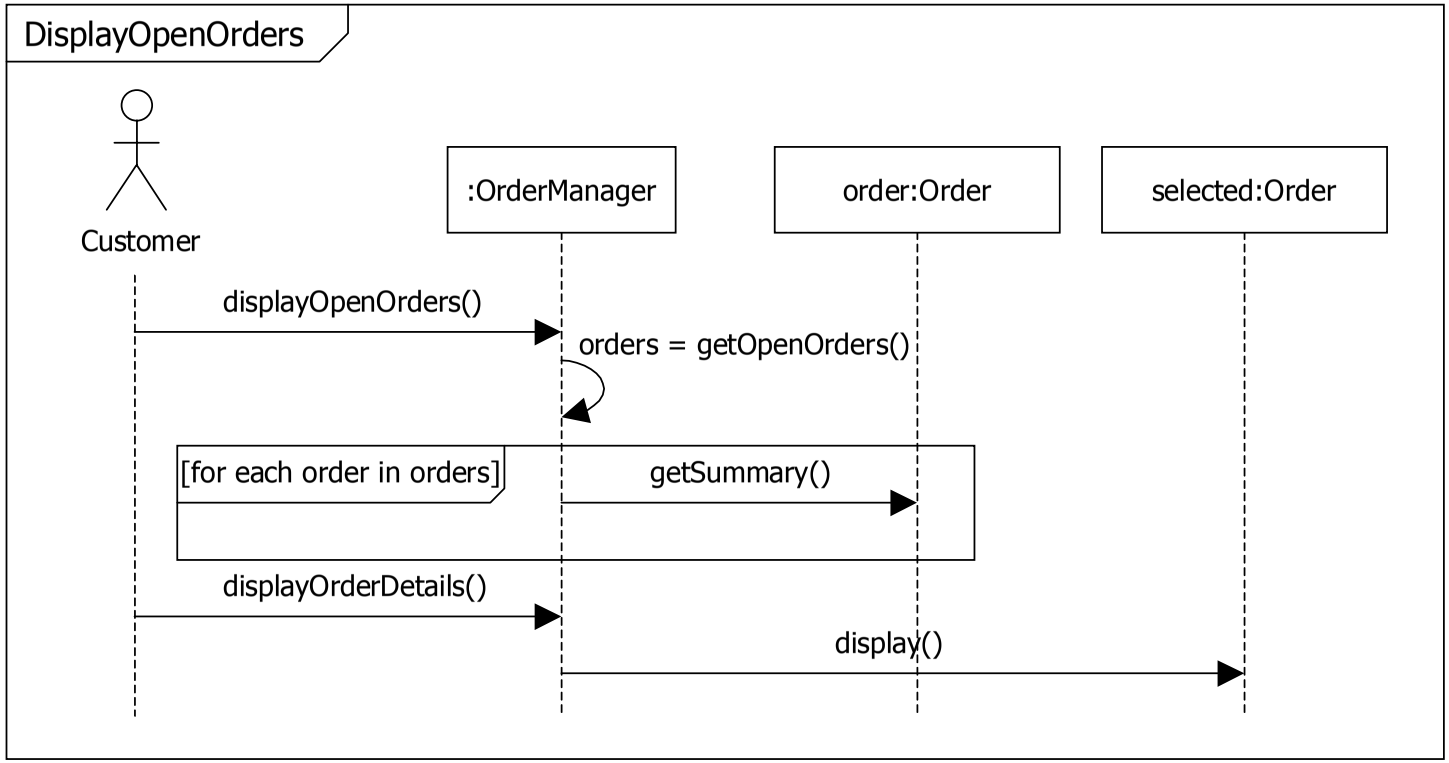
6.2.13 DeleteUser Sequence Diagram



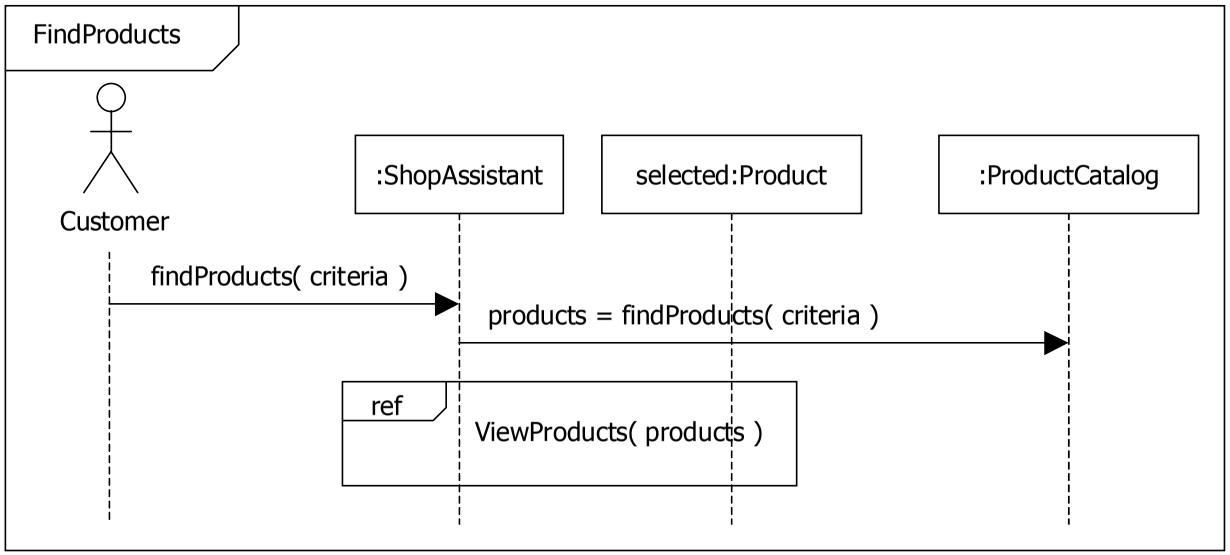
6.2.14 DisplayBasket Sequence Diagram



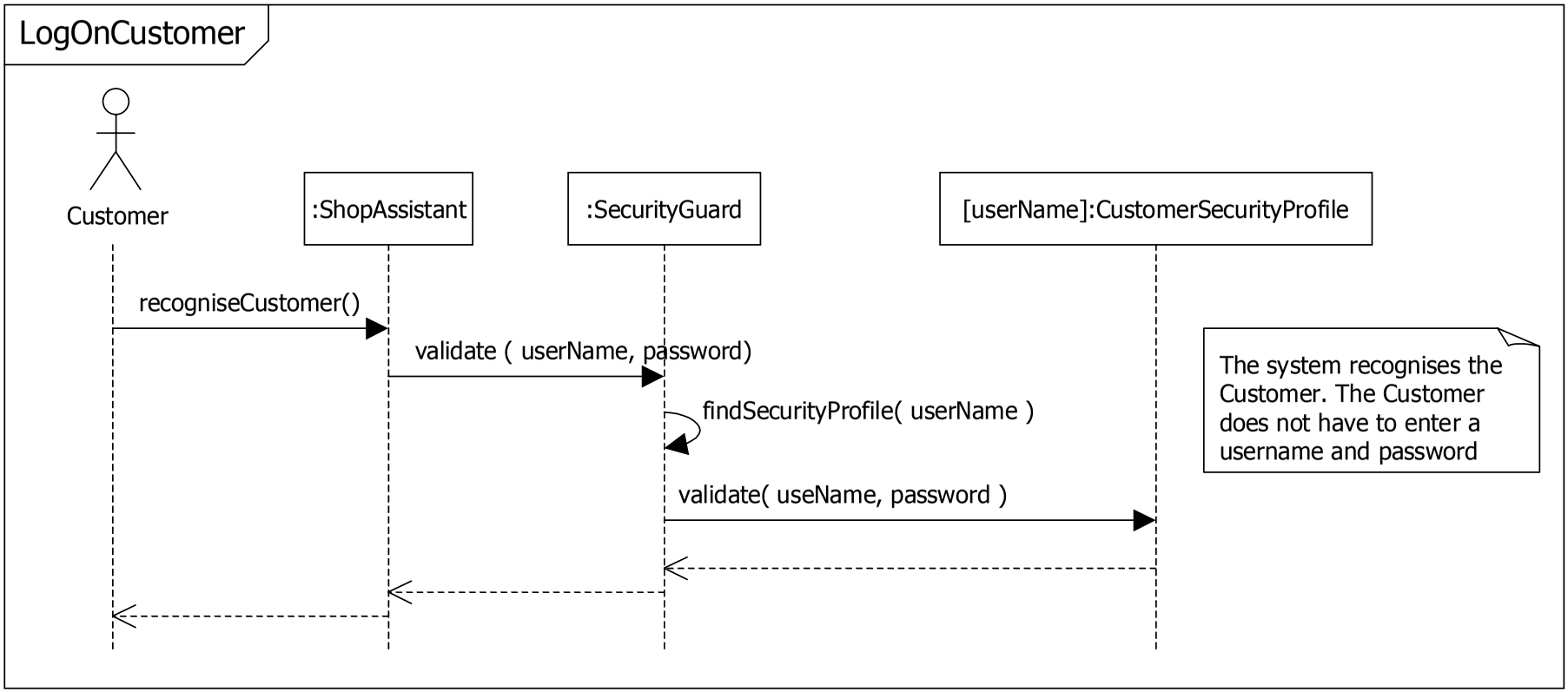
6.2.15 DisplayOpenOrders Sequence Diagram



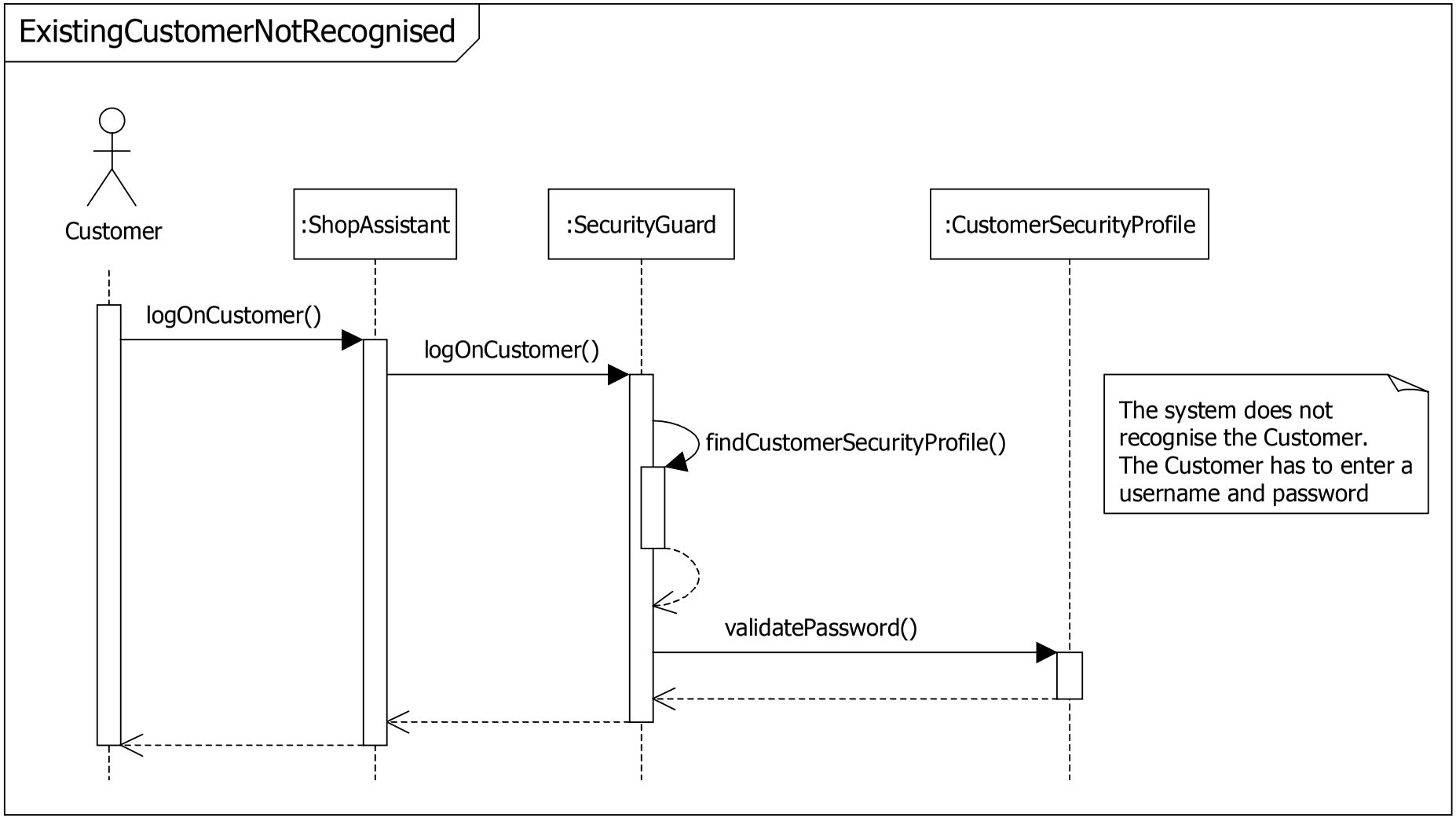
6.2.16 FindProducts Sequence Diagram



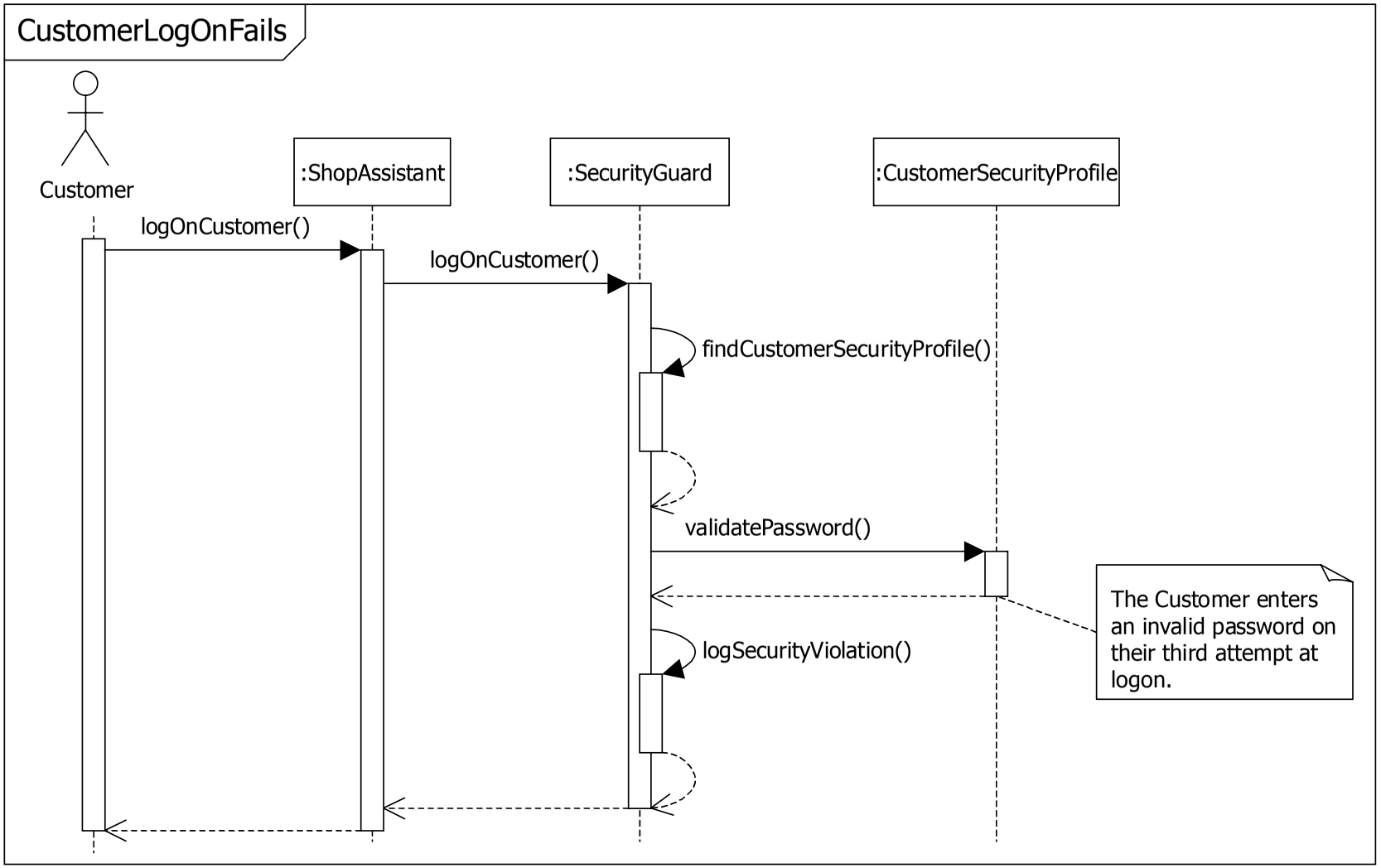
6.2.17 LogOnCustomer Sequence Diagram



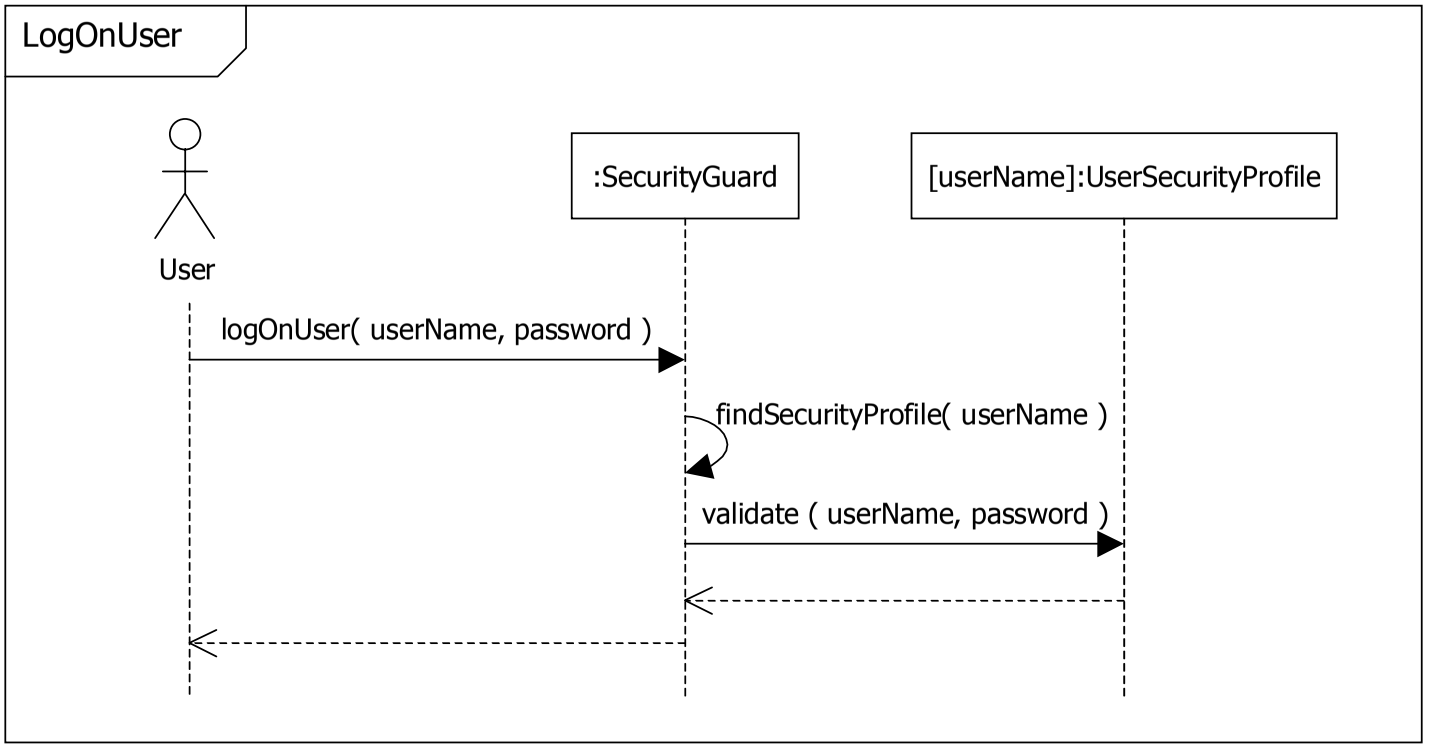
6.2.18 LogonCustomer Sequence - ExistingCustomerNotRecognised



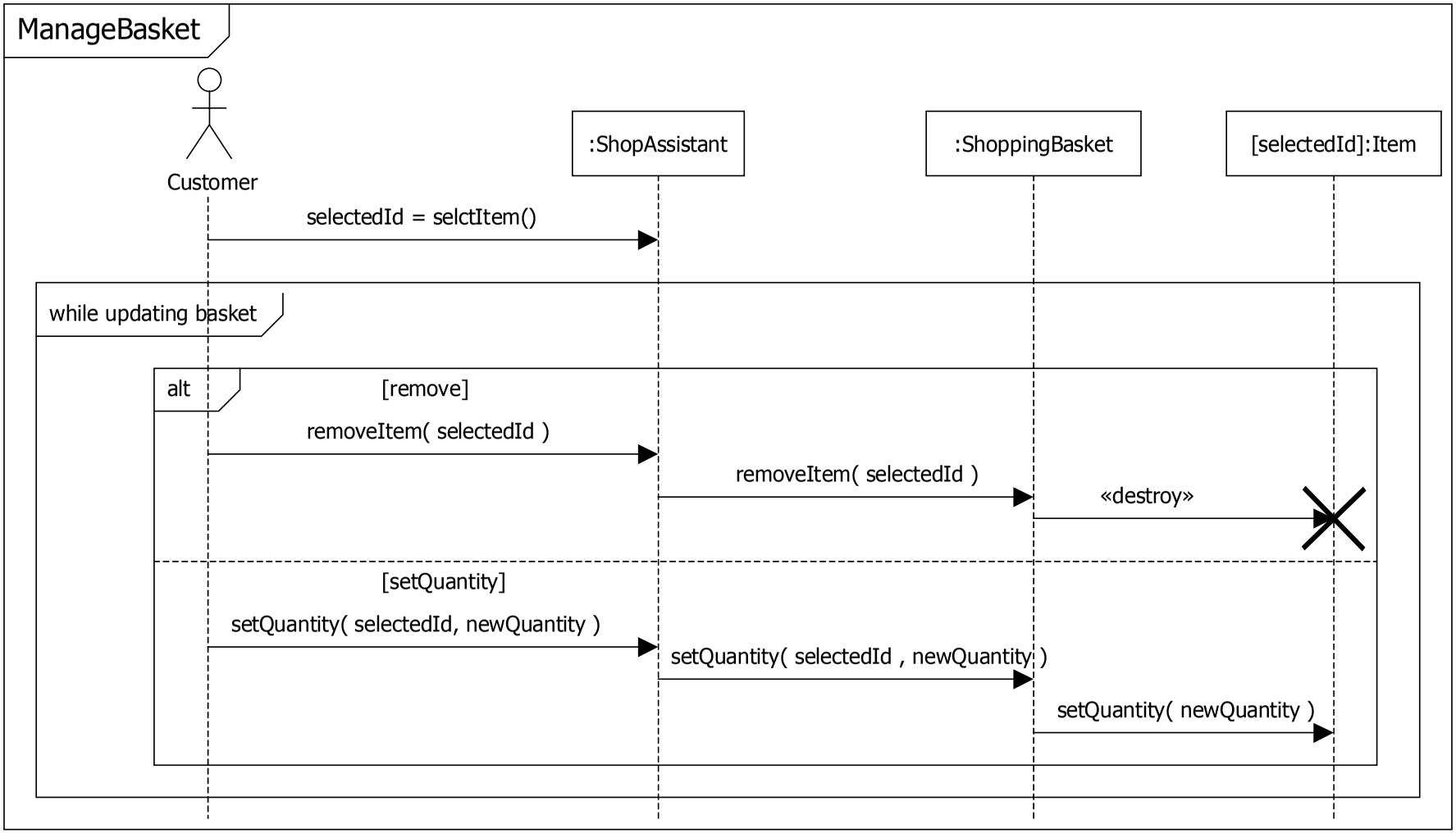
6.2.19 LogonCustomer Sequence - CustomerLogOnFails



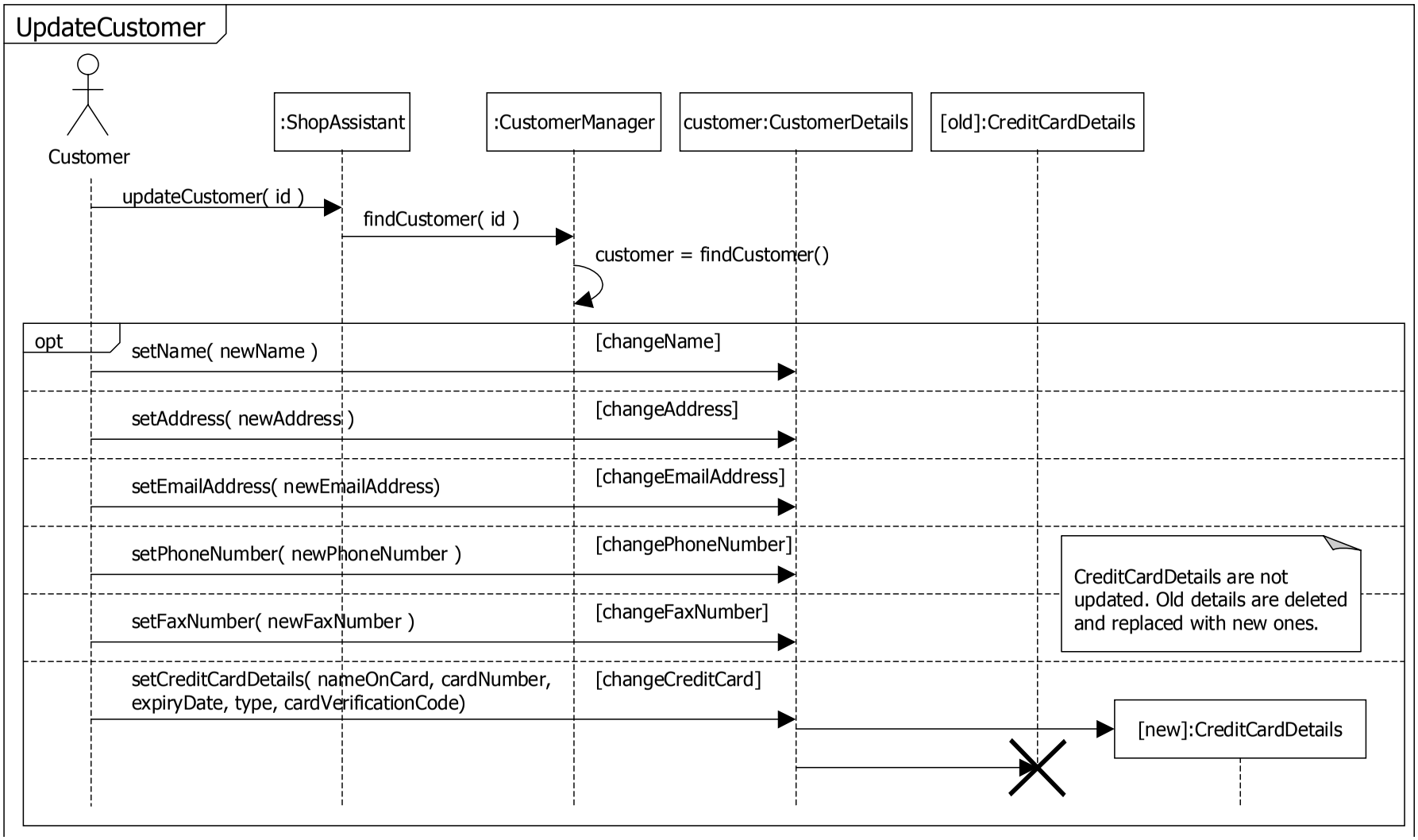
6.2.20 LogOnUser Sequence Diagram



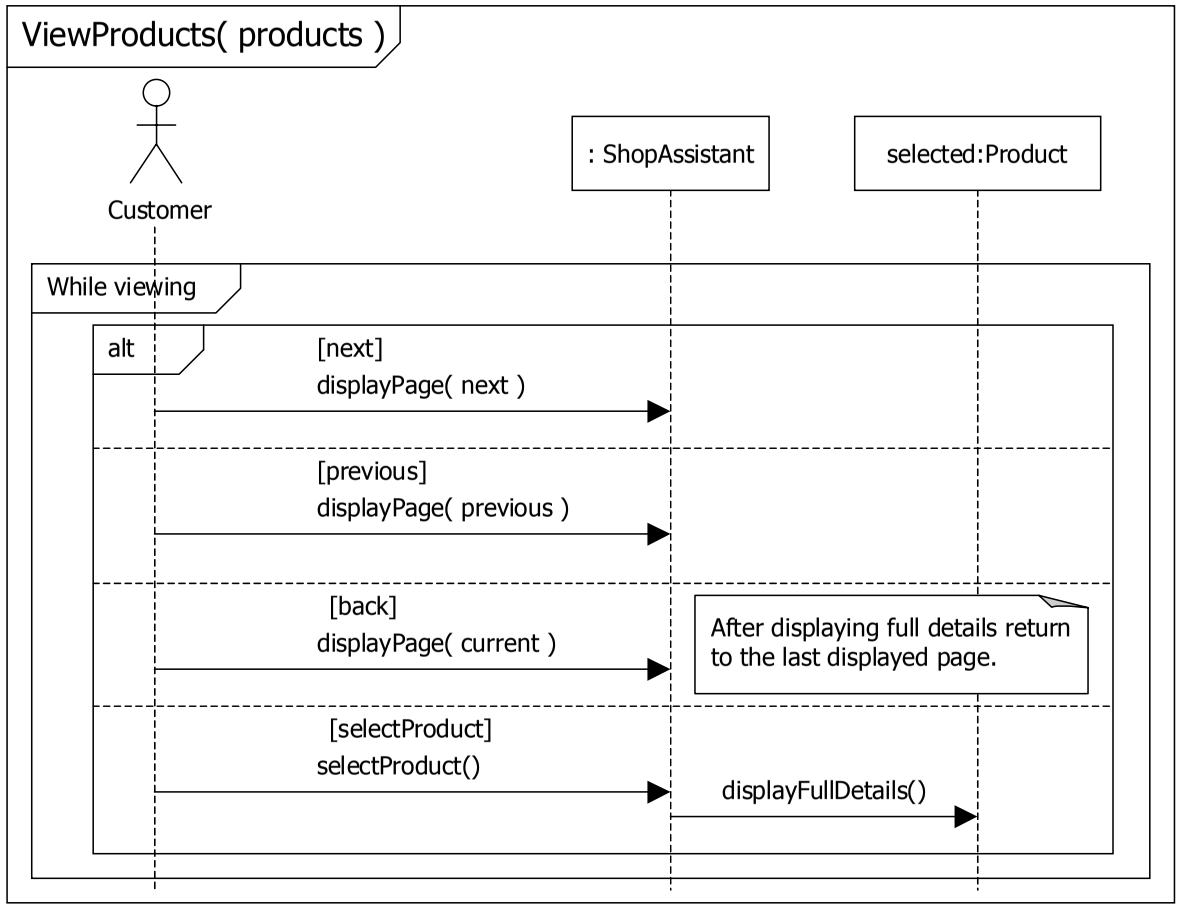
6.2.21 ManageBasket Sequence Diagram



6.2.22 UpdateCustomer Sequence Diagram



6.2.23 ViewProducts Sequence Diagram



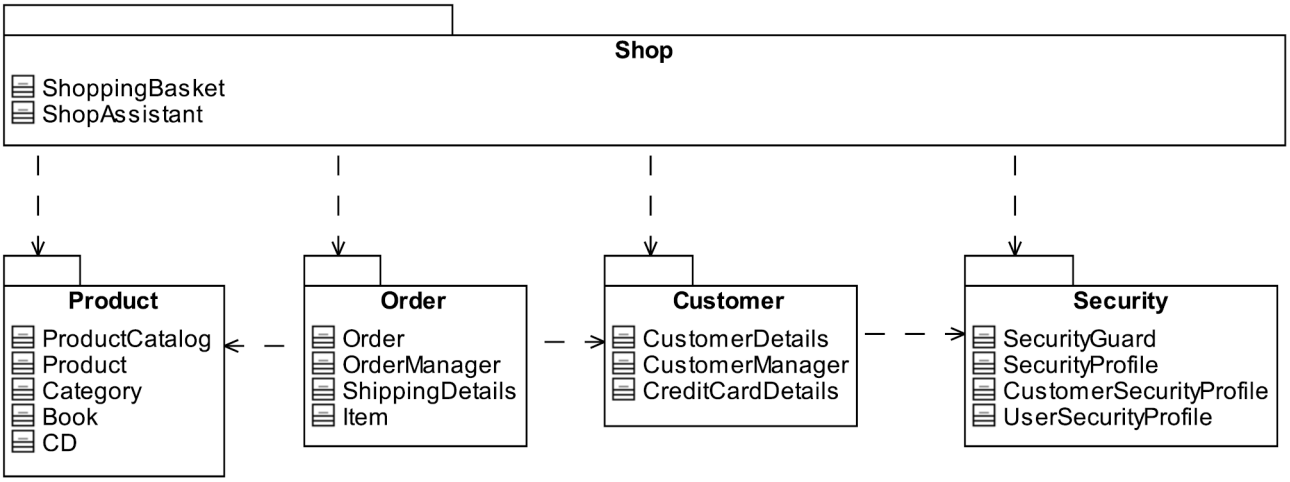
7 ECP analysis class diagrams

7.1 Introduction

This section contains the final ECP analysis class diagrams that resulted from the process of use case realization.

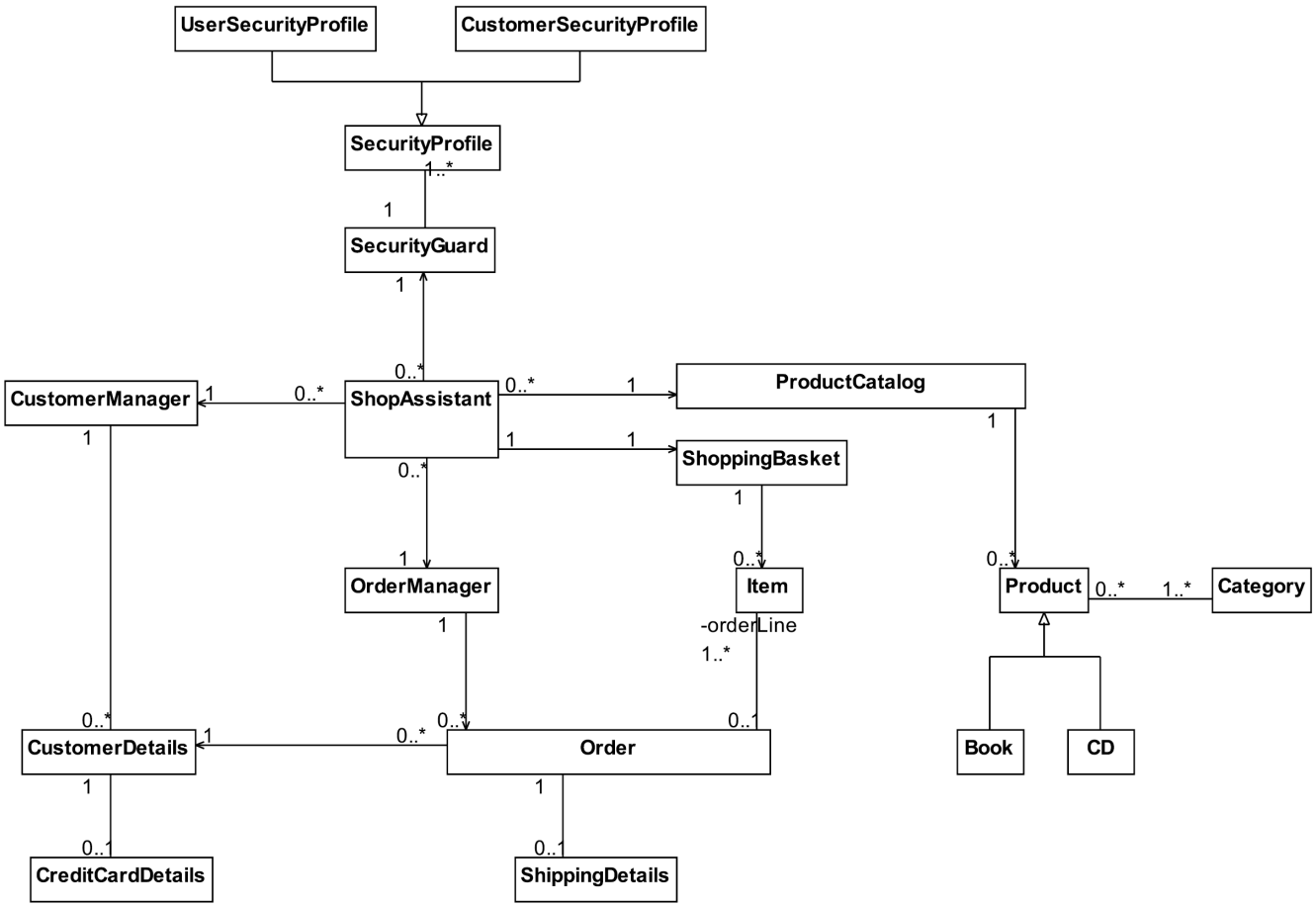
7.2 ECP analysis package diagram

Figure 5:



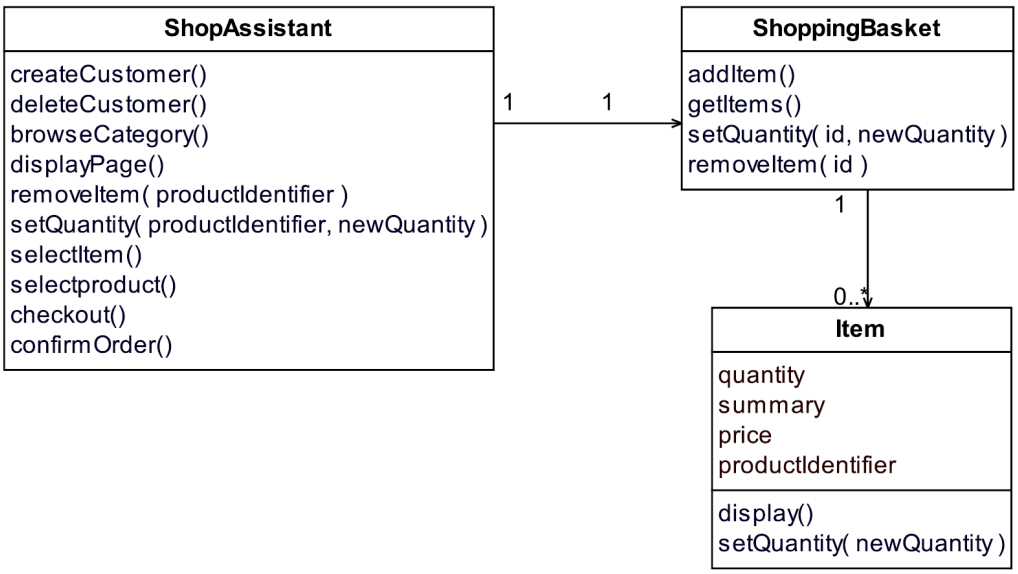
7.3 Overview class diagram

Figure 6:



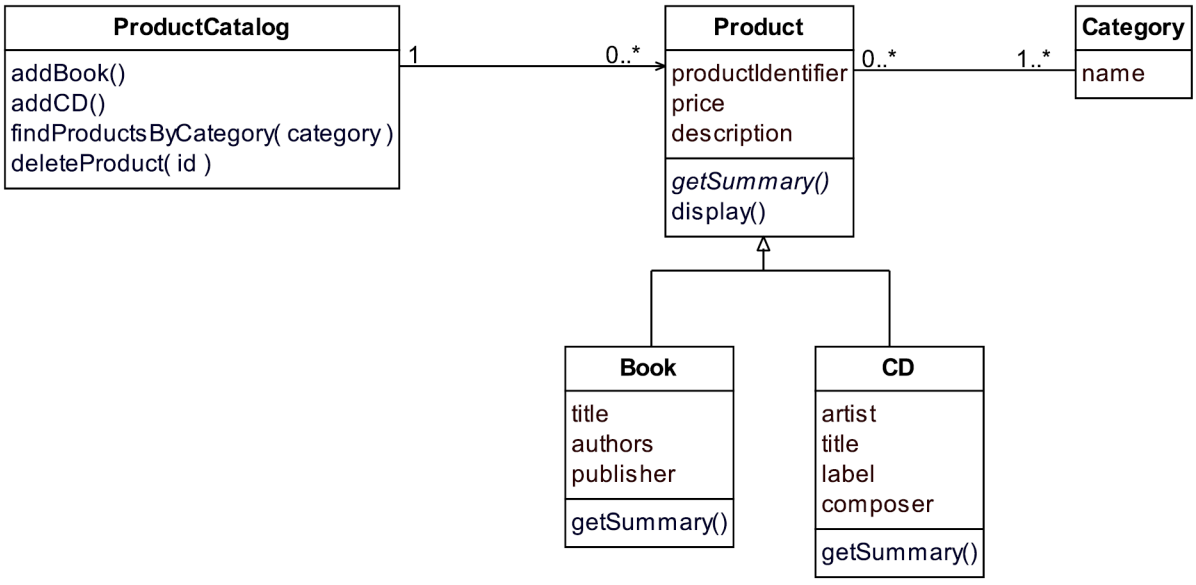
7.4 The Shop package

Figure 7:



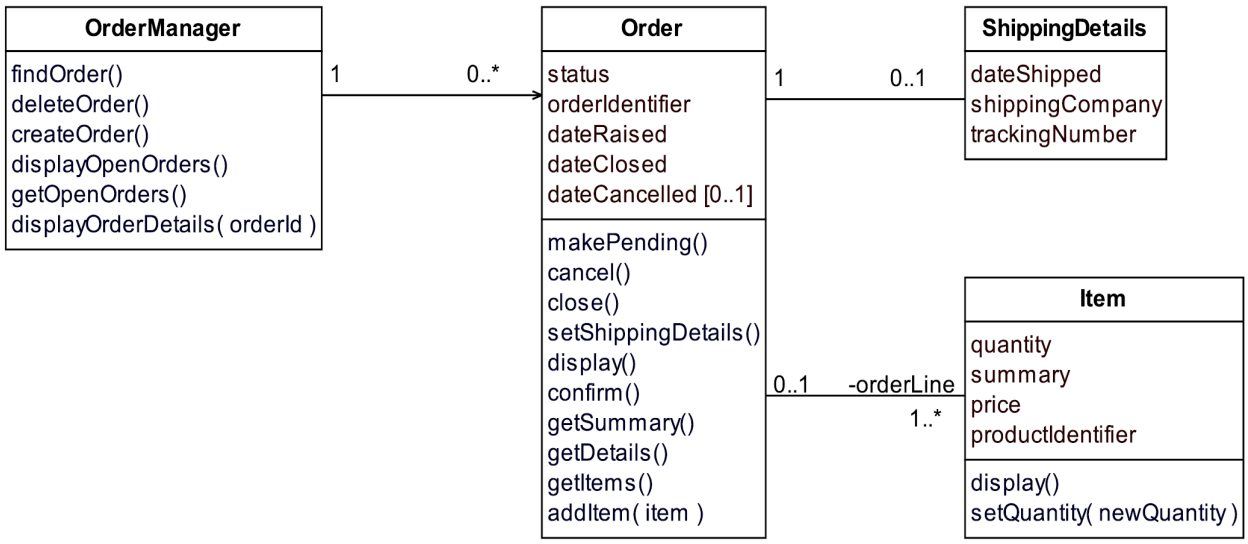
7.5 The Product package

Figure 8:



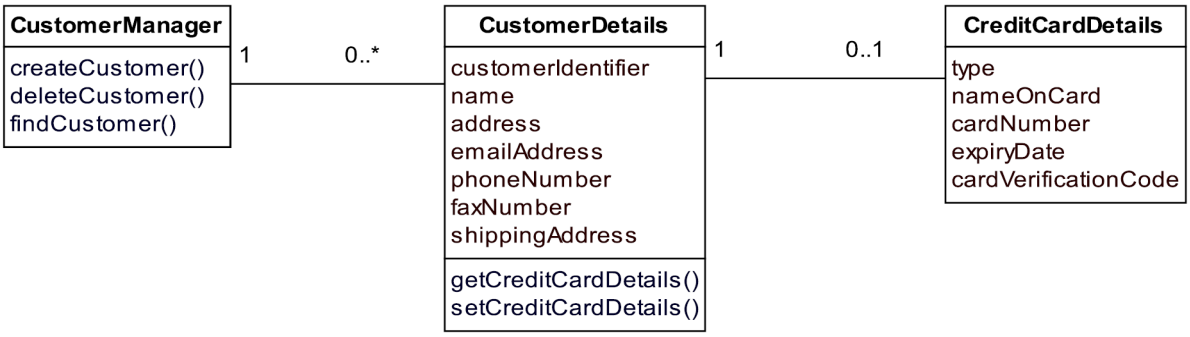
7.6 The Order package

Figure 9:

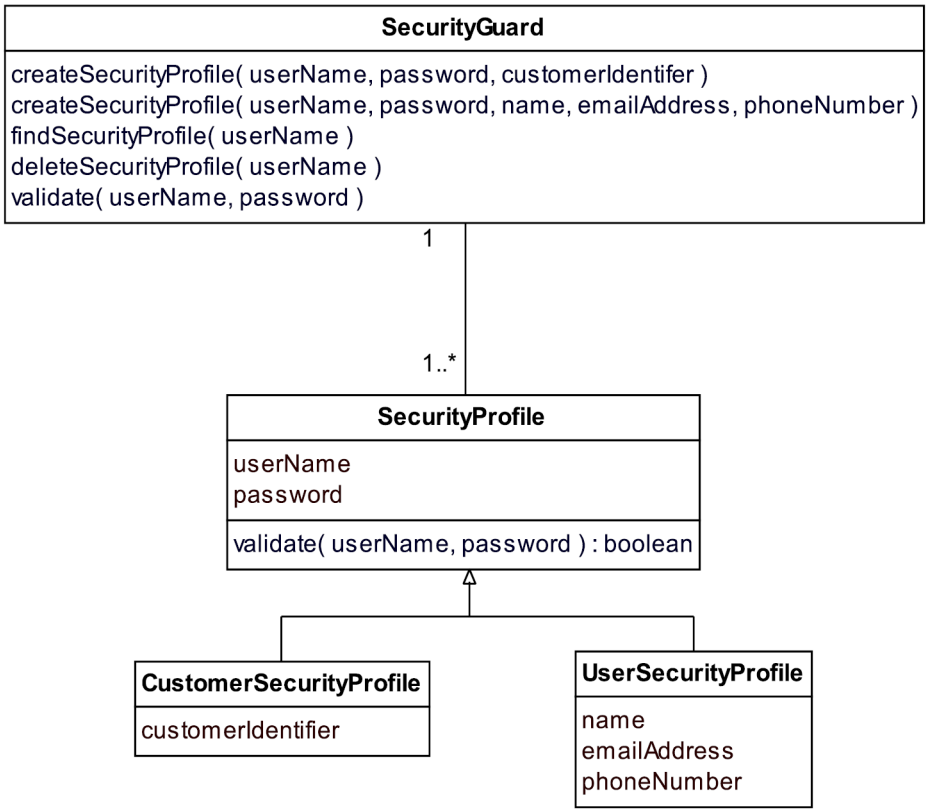


7.7 The Customer package

Figure 10:



7.8 The Security package



8 ECP design model

8.1 Introduction

In iteration 1, we implement the use cases AddProductToCatalog, RemoveProductFrom-Catalog for books.

In this iteration we develop a vertical slice down through the system from the user interface layer to the database access layer.

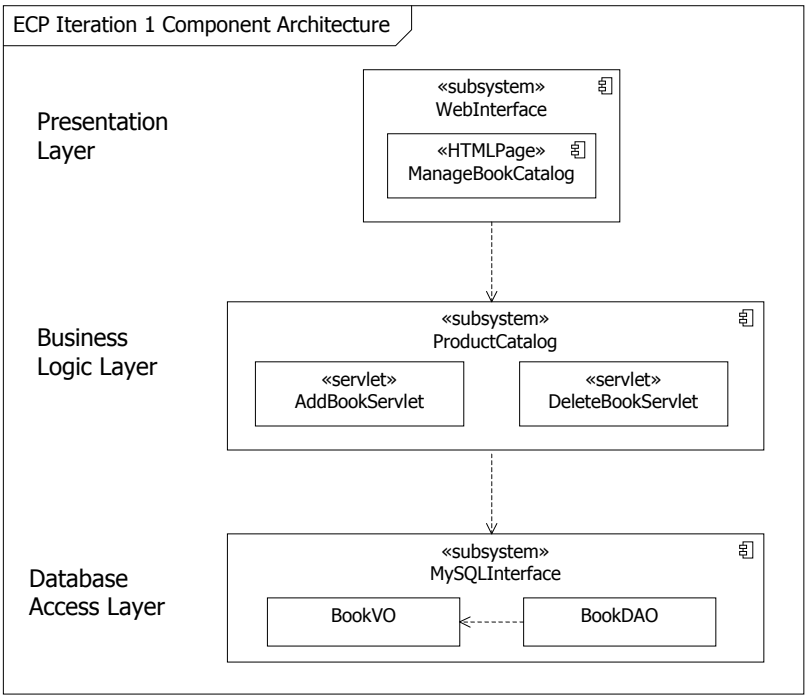
8.2 Subsystems

The component diagram in Figure 11 shows the layered architecture for the ECP system. These layers are:

- Presentation - the user interface layer;
- Business Logic - the business logic of the application;
- Database Access - the interface to the database.

Each of these layers contains one or more components stereotyped «subsystem». Each of these components represents a logical unit of decomposition of the system. Each component is realized by one or more classes

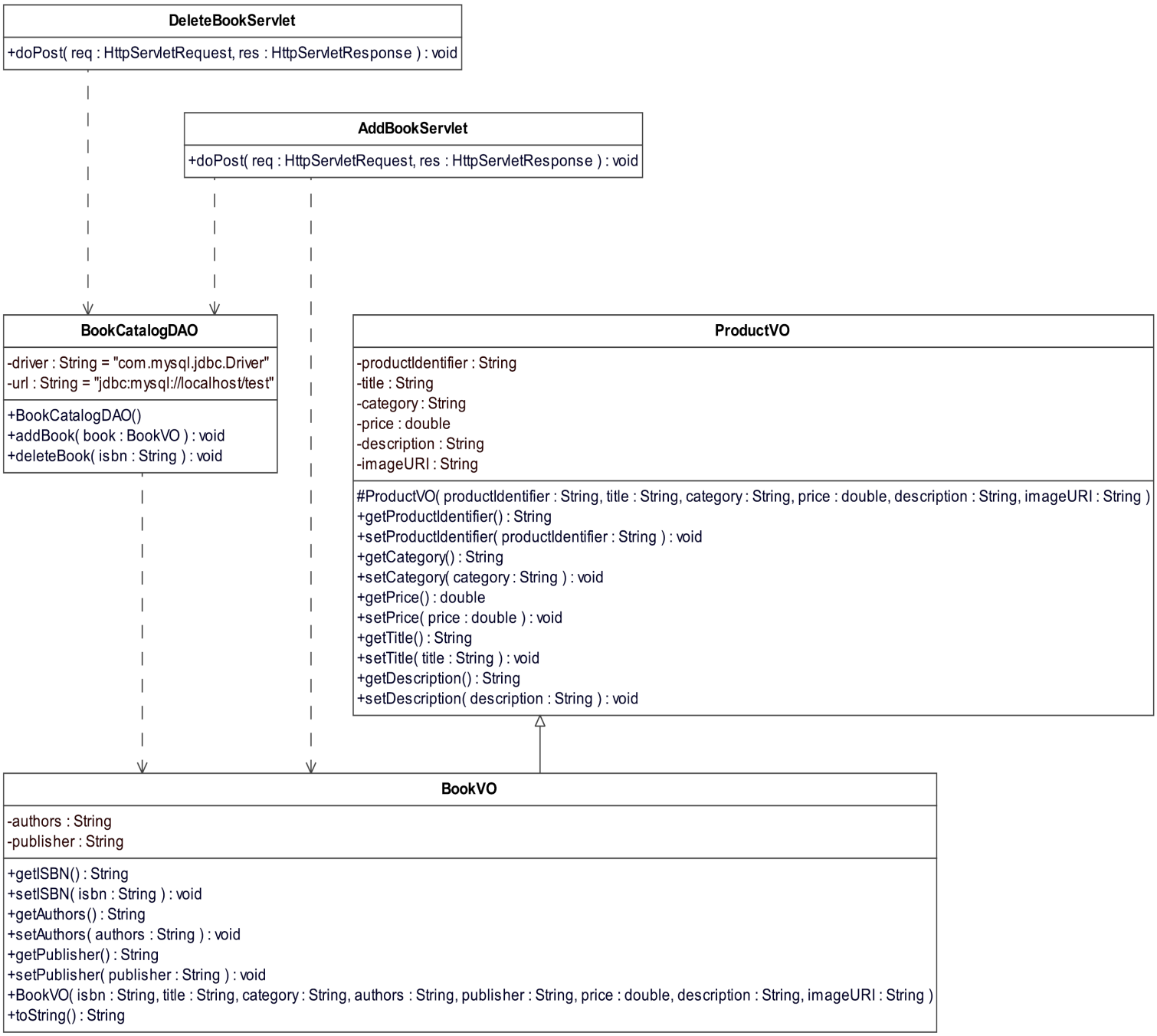
Figure 11:



8.3 Design class diagram

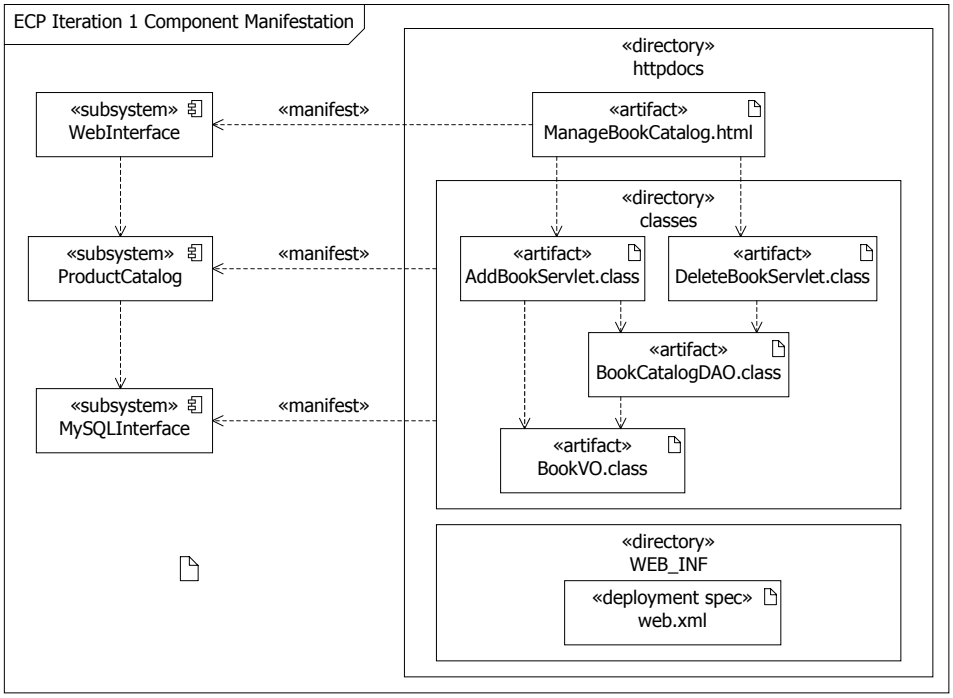
This class diagram shows how the Business Logic layer (AddBookServlet and DeleteBookServlet) interacts with the Database Access layer (BookCatalogDAO and BookVO) to add and delete book entries from the database.

Figure 12:



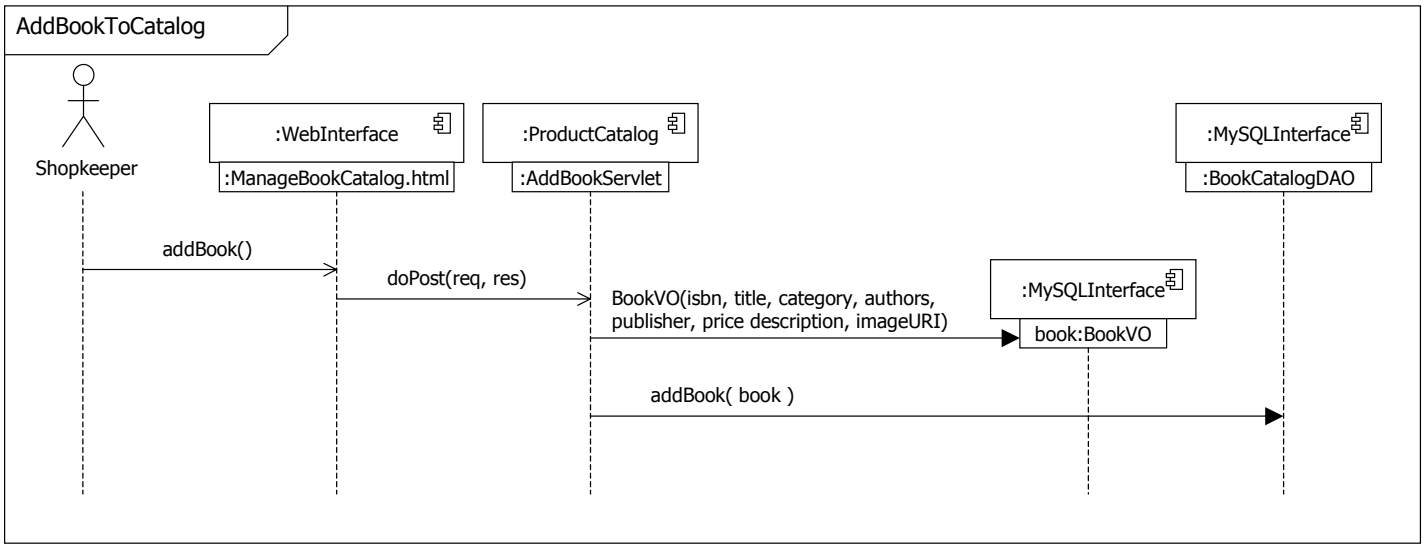
8.4 Artifacts

Figure 13:



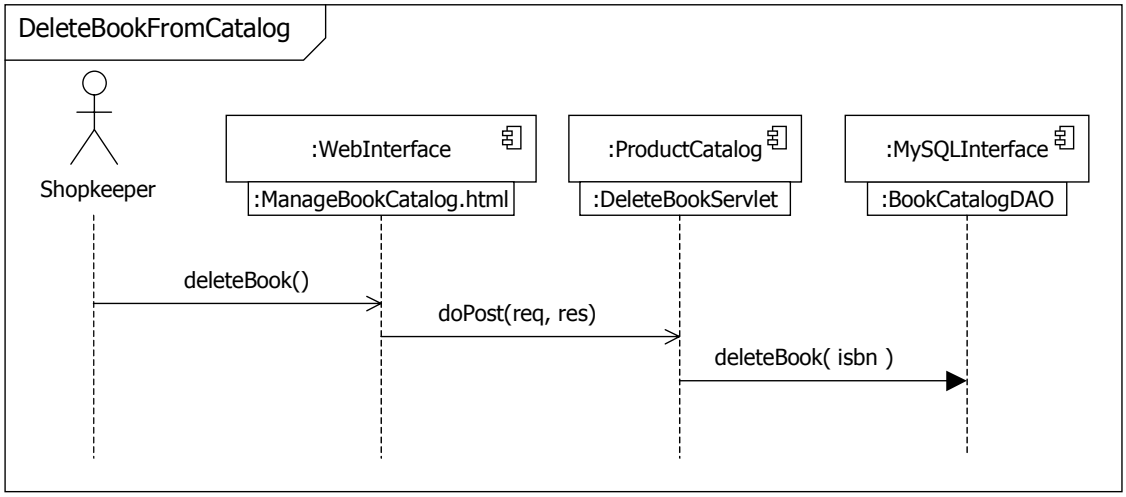
8.5 AddBookToCatalog design sequence diagram

Figure 14:



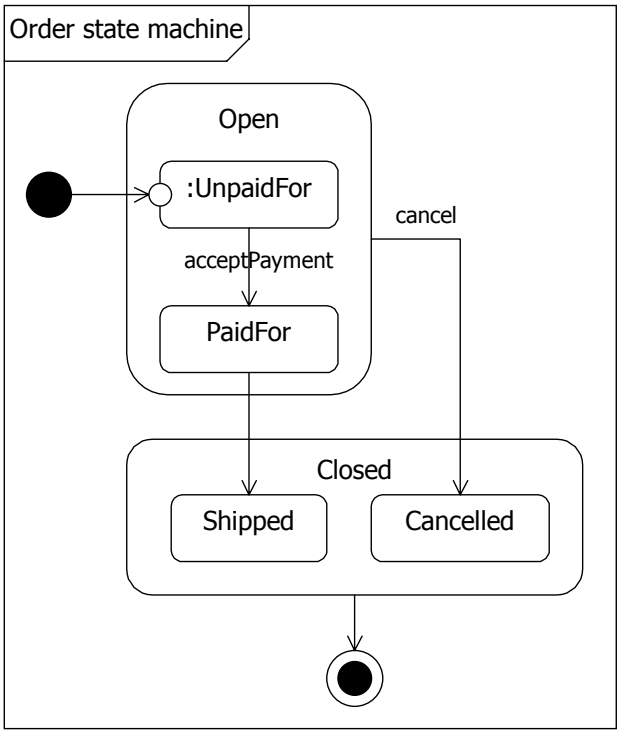
8.6 DeleteBookFromCatalog design sequence diagram

Figure 15:



8.7 Order state machine

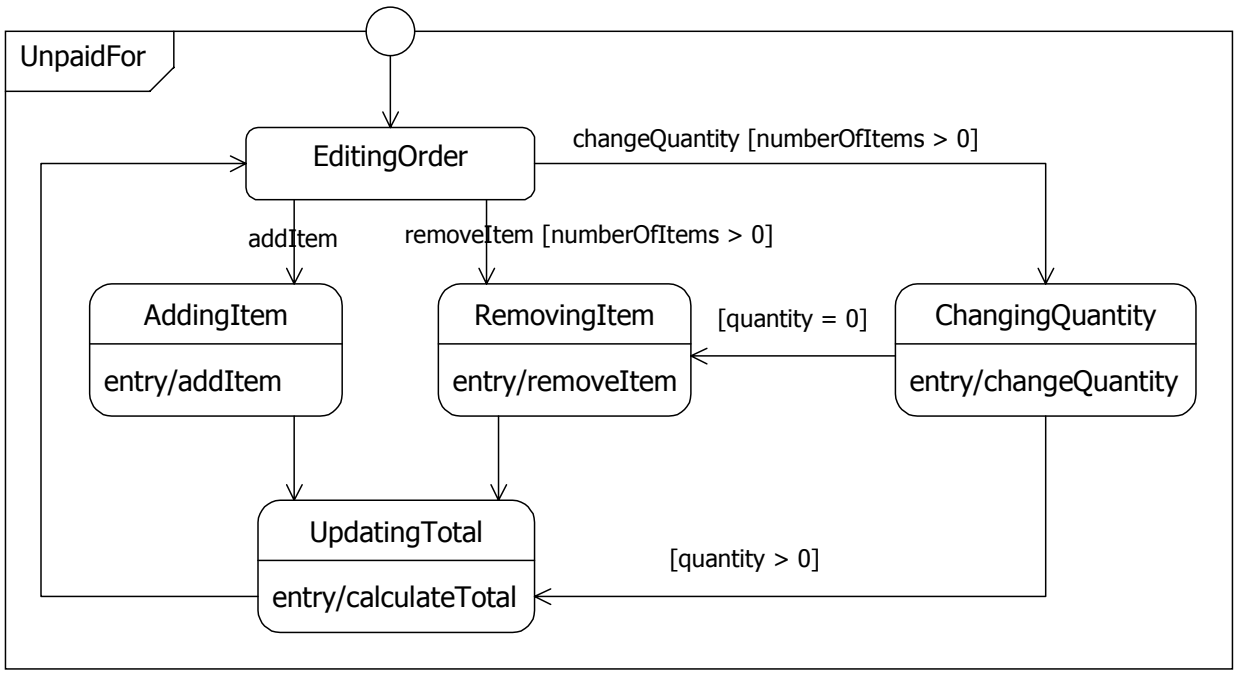
Figure 16:



8.8 UnpaidFor submachine state - adding and removing items

Here is the submachine for the UnpaidFor state.

Figure 17:



9 ECP deployment model

9.1 Deployment diagram

Figure 18: