

Shopping Cart Application

a project for

COP 5339 - Object-Oriented Software Design
Dr. Ionut Cardei

Group 17

Newman Souza

Seth Moore

Functional Specification For The Shopping Cart Application

The Shopping Cart Application allows a seller to maintain an inventory of items for sale and allows customers to browse and purchase those items.

Application Features

The Shopping Cart Application allows a seller to maintain an inventory of items available for sale and customers to browse the items, add items to their cart, and purchase the contents of their cart. The Shopping Cart Application manages the changes to the seller's costs, revenues, profits, and inventory state as customers purchase items and the seller updates the inventory.

The application provides the following features:

1. Users can log in.
2. Users can log out.
3. Customers can browse products available for purchase.
4. Customers can review product details.
5. Customers can add items to their shopping cart.
6. Customers can review/update the contents of their shopping cart.
7. Customers can pay for the items in their cart.
8. The seller can add new products to the inventory, providing
 - ID
 - Name
 - Full description
 - Type/category
 - Quantity
 - Invoice price
 - Sell price
9. The seller can review/update the inventory.
10. The seller can remove products from inventory.
11. The seller can view current state of inventory including costs, revenues and profit.

The application maintains a persistent inventory and financial state between executions.

User Interface

The Shopping Cart Application has a graphical user interface with the following windows displayed as appropriate:

1. Login window
 - Contains a text box for entering username.
 - Contains a text box for entering password.
 - Contains a Login button.
2. Browse Products window (Customer)
 - Contains a list of available products (name, price, quantity available).
 - Contains a component that shows a summary of the shopping cart.
 - Contains a *Add to Cart* button for each product
 - Contains a *Checkout* button.
 - Contains a *Logout* button.
3. Browse Inventory window (Seller)
 - Contains a list of all products (name, invoice price, sell price, quantity available).
 - Contains a component that shows a summary of seller's financials (costs, revenues, profits).
 - Contains an *Add Product* button.
 - Contains a *Logout* button.
4. Review Details window (Customer)
 - Contains the product's detailed information (ID, name, full description, sell price, and quantity available).
 - Contains an *OK* button.
5. Edit Product window (Seller)
 - Contains the product's detailed information (ID, name, full description, invoice price, sell price, and quantity available), and all fields are modifiable.
 - Contains *Update*, *Delete* and *Cancel* buttons.
6. Checkout window
 - Contains a list of products in the shopping cart (name, price, quantity).
 - Contains Increment and Decrement buttons for each product.
 - Contains a payment form.
 - Contains a *Pay* button.

Application Architecture

The Shopping Cart Application is implemented as a Java application with Swing GUI and is executed from the command-prompt terminal.

Use Cases For The Shopping Cart Application

Startup

1. User starts the application.
2. CartSystem creates UI.
3. UI displays LoginScreen.

Log In

1. User carries out Startup.
2. User enters a username and a password.
3. User clicks the *Login* button.
4. UI sends the username and password to CartSystem.
5. CartSystem validates username and password.
6. CartSystem determines that the User is a customer.
7. CartSystem loads inventory.
8. UI displays CustomerScreen.
9. Customer browses products.

Variation #1. Login matches Seller

- 1.1 Start at Step 5.
- 1.2 CartSystem determines that the user is a seller.
- 1.3 CartSystem loads inventory
- 1.4 UI displays SellerScreen.
- 1.5 Seller browses Inventory.

Variation #2. Validation fails.

- 2.1 Start at Step 4.
- 2.2 CartSystem does not validate username or password.
- 2.3 UI displays a failed login message.
- 2.4 User clicks the *OK* button.
- 2.5 UI displays LoginScreen.

Review Product Details

1. Customer clicks on a product.
2. UI displays the product's detailed information (ID, name, description, price, and quantity available)
3. Customer clicks the *OK* button.
4. UI displays CustomerScreen.

Add Product to Cart

1. Customer clicks the *Add to Cart* button for a specific product.
2. CartSystem adds 1 of the product to Cart.
3. CartSystem updates Cart summary.
4. CartSystem updates quantity and financials in Inventory.

Checkout

1. Customer clicks the *Checkout* button.
2. UI displays the list of products in the shopping cart (name, price, and quantity).
3. UI displays Increment and Decrement buttons for each product.
4. UI displays the Cart summary, a payment form and a Pay button.

Variation #1. Increment Product

- 1.1 Start at Step 4.
- 1.2 Customer clicks the *Increment* button of a product.
- 1.3 CartSystem updates Cart totals.
- 1.4 CartSystem updates quantity and financials in Inventory.
- 1.5 Go to Step 3.

Variation #2. Decrement Product

- 1.1 Start at Step 4.
- 1.2 Customer clicks the *Decrement* button of a product.
- 1.3 CartSystem updates Cart totals.
- 1.4 CartSystem updates quantity and financials in Inventory.
- 1.5 Go to Step 3.

Pay

1. Customer carries out Checkout.
2. Customer enters payment information on form.
3. Customer clicks the *Pay* button.
4. CartSystem sends transaction details to PaymentValidator.
5. PaymentValidator approves transaction.
6. CartSystem updates database.
7. UI clears the shopping cart.
8. UI displays a message confirming the sale.
9. Customer clicks the *OK* button.
10. UI displays CustomerScreen.

Add New Product

1. Seller clicks the *Add Product* button.
2. UI displays a form for product information (ID, name, description, invoice price, sell price, and quantity available), a *Cancel* button and a *Save* button.
3. Seller enters product information.
4. Seller clicks the *Save* button.
5. CartSystem updates inventory information.
6. CartSystem updates seller's financials.
7. CartSystem updates database.
8. UI displays SellerScreen.

Variation #1. Cancel Add

- 1.1 Start at Step 4.
- 1.2 Seller clicks the *Cancel* button.
- 1.3 UI displays SellerScreen.

Update Product

1. Seller clicks on a product.
2. UI displays the product's detailed information (ID, name, full description, invoice price, sell price, and quantity available)
3. Seller updates product information.
4. Seller clicks the *Update* button.
5. CartSystem updates inventory information.
6. CartSystem updates seller's financials.
7. CartSystem updates database.
8. UI displays SellerScreen.

Variation #1. Delete Product

1. Start at Step 3.
2. Seller clicks the *Delete* button.
3. UI displays a message asking for confirmation.
4. Seller clicks the Yes button.
5. CartSystem updates inventory information.
6. CartSystem updates database.
7. UI displays SellerScreen.

Variation #2. Cancel Update

- 1.1 Start at Step 3.
- 1.2 Seller clicks the *Cancel* button.
- 1.3 UI displays SellerScreen.

Logout

1. User clicks the *Logout* button.
2. The UI clears the Cart.
3. UI displays LoginScreen.

CRC Cards

CartSystem

- Responsibilities
 - Login process
 - User validation process
 - Manage DBManager
 - Communicate with PaymentValidator
- Collaborators
 - UI
 - DBManager
 - UserList
 - PaymentValidator
 - Inventory

UI

- Responsibilities
 - Manage interaction with user
 - Manage Cart
 - Manage Inventory
 - Display screens
 - Receive input
- Collaborators
 - CartSystem
 - Inventory
 - Cart
 - Product

DBManager

- Responsibilities
 - Load data
 - Save data
- Collaborators
 - Inventory
 - UserList

PaymentValidator

- Responsibilities
 - Validate payments
- Collaborators

Inventory

- Responsibilities
 - Manage list of products in Inventory
 - Manage Seller's financials
- Collaborators
 - Product

Cart

- Responsibilities
 - Manage list of products in Cart
 - Provide access to Cart totals
- Collaborators
 - Product

Product

- Responsibilities
 - Manage state
- Collaborators

UserList

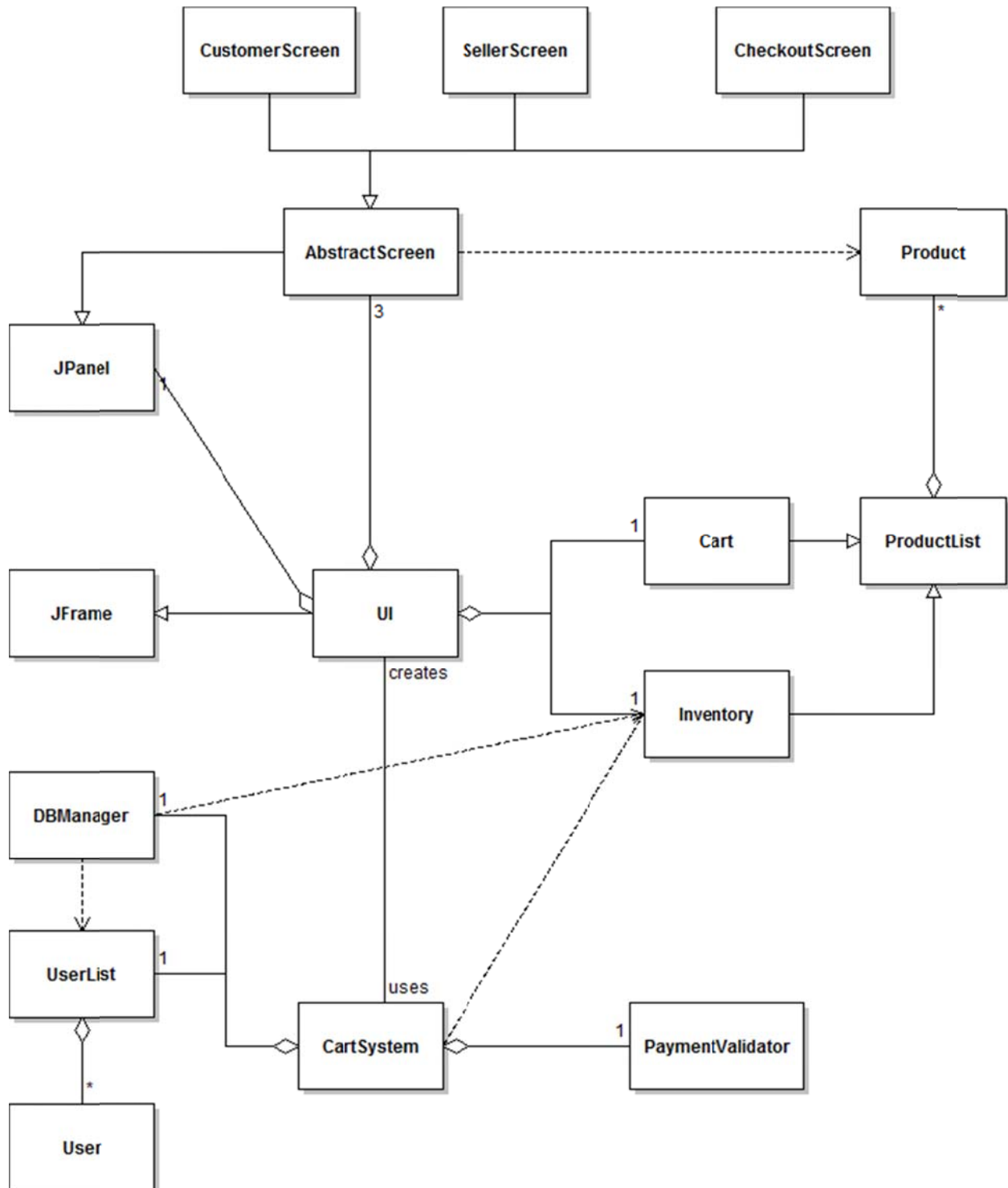
- Responsibilities
 - Manage list of Users
 - Validate User
- Collaborators
 - User

User

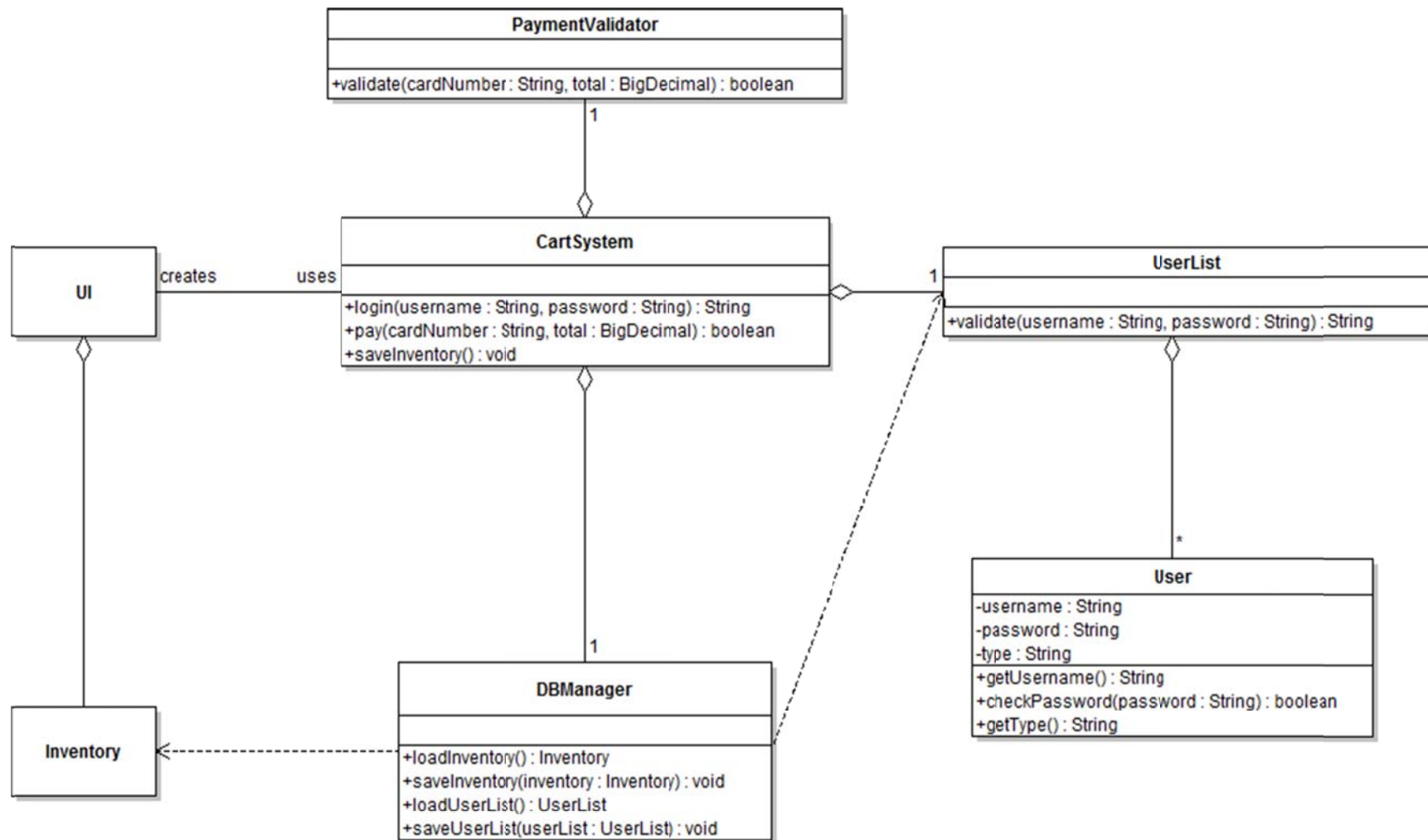
- Responsibilities
 - Manage User state
- Collaborators

Class Diagrams

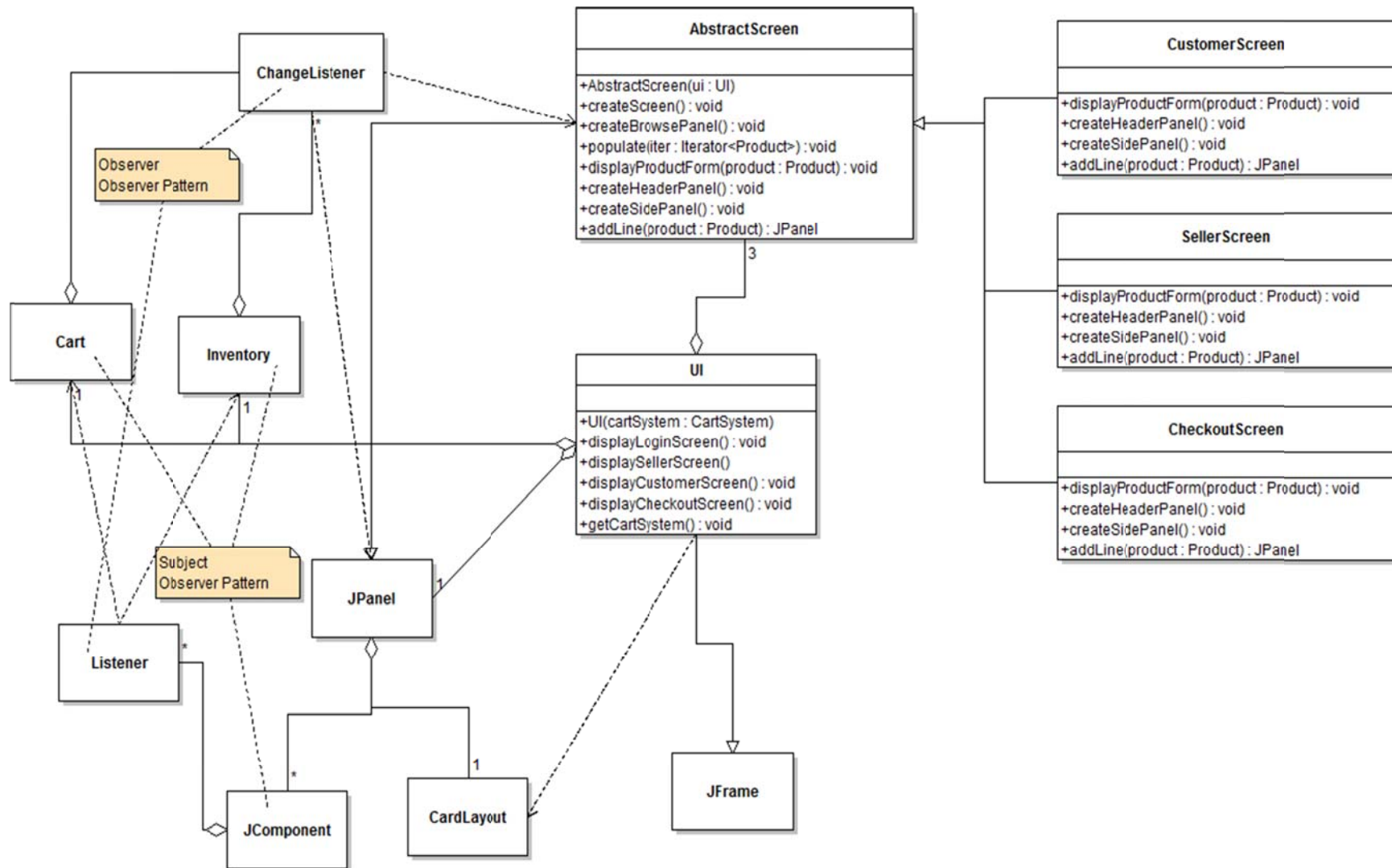
Overview



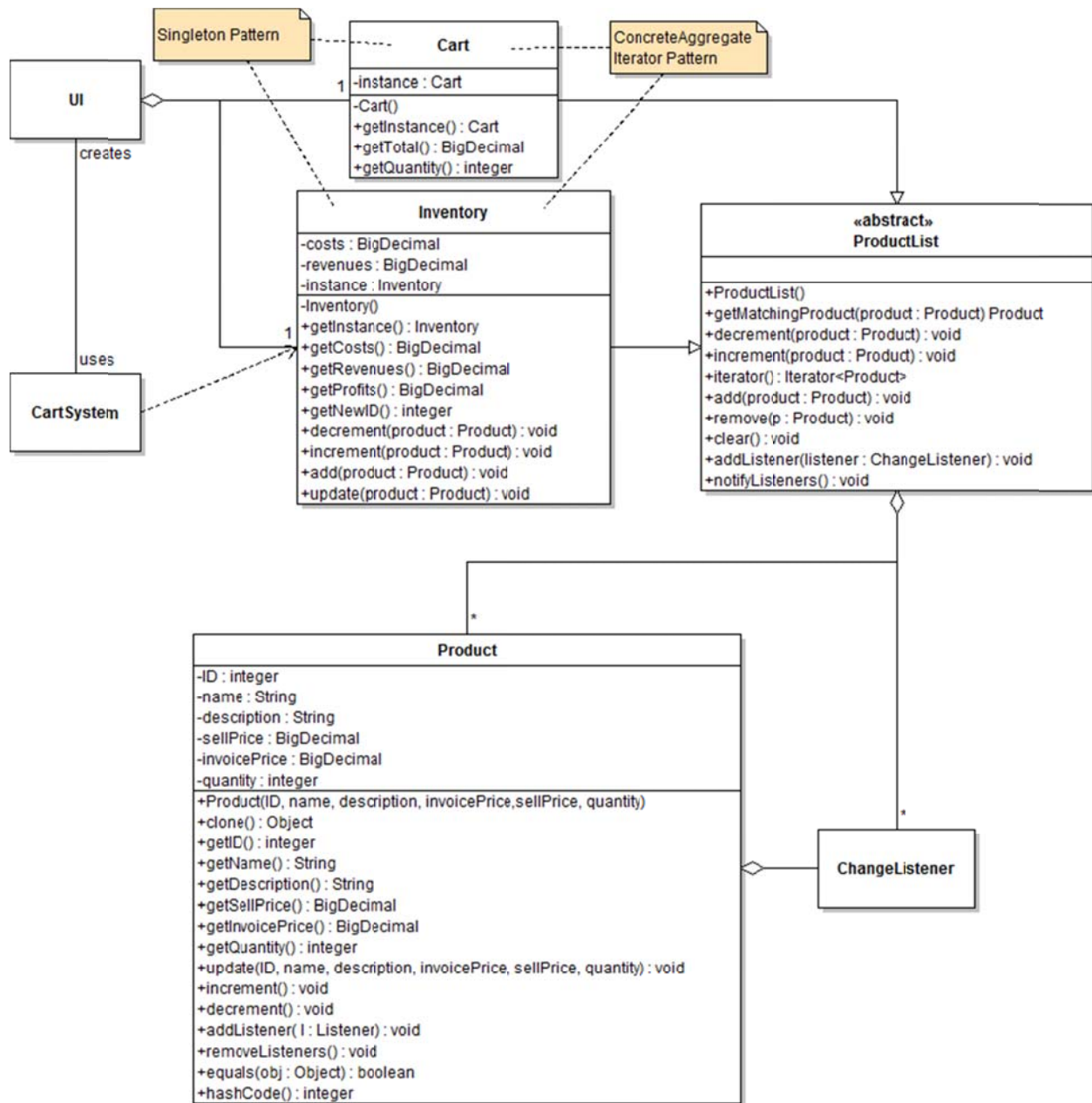
CartSystem Detail



UI Detail

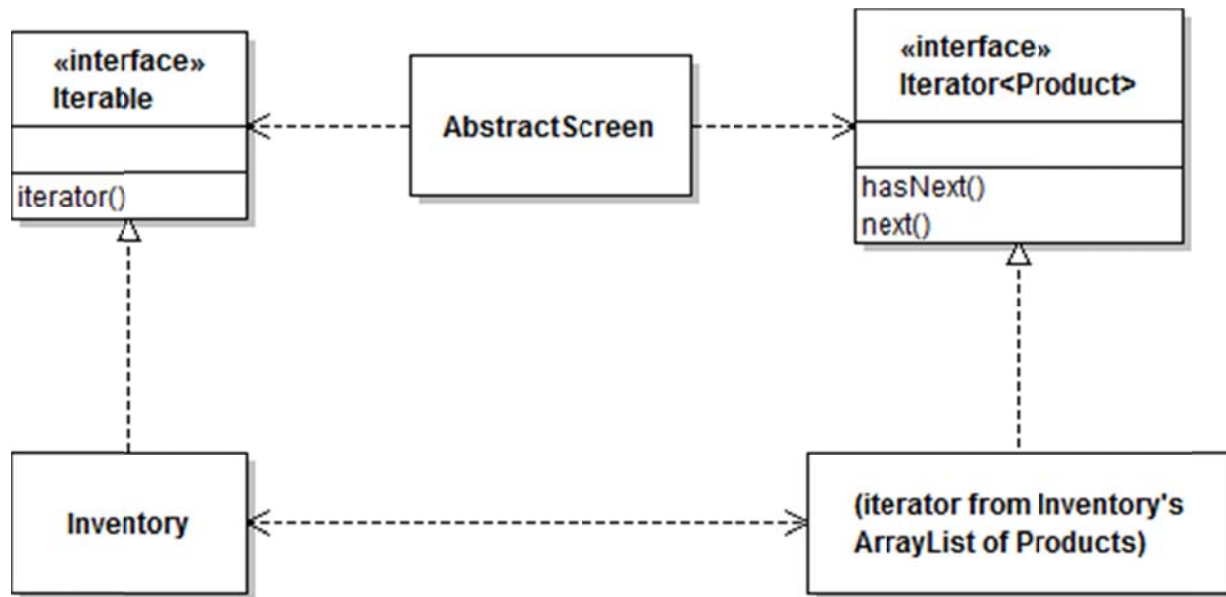


ProductList Detail



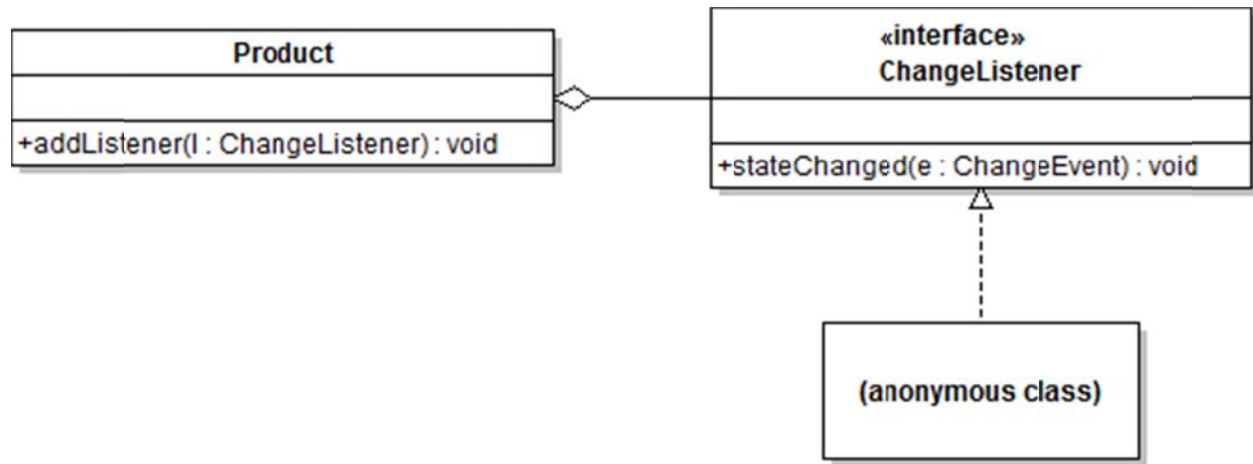
Patterns Used

Iterator Pattern



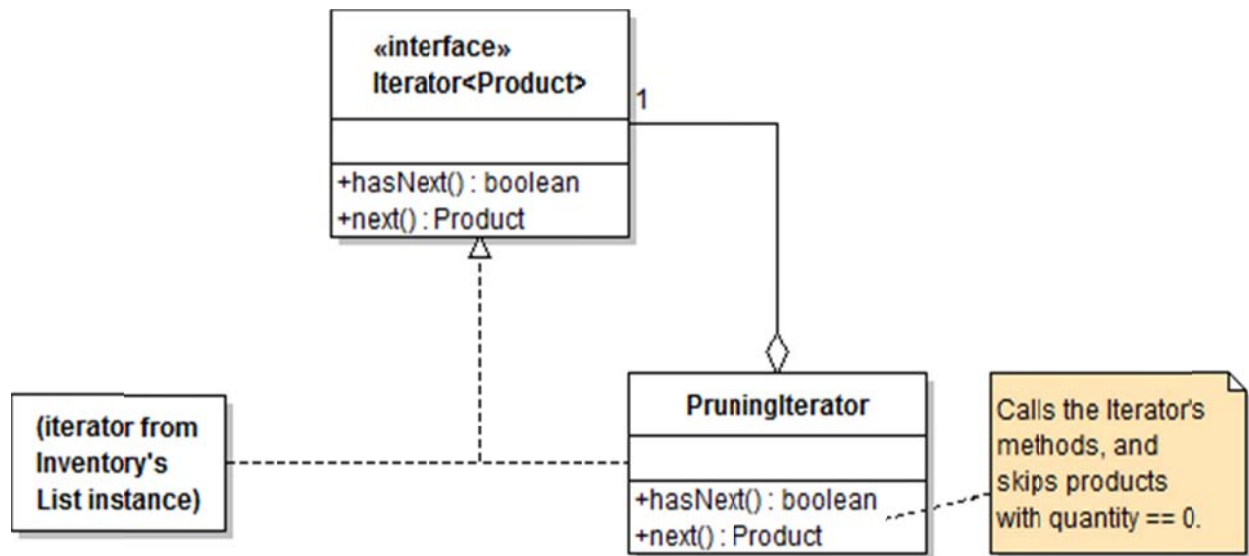
Name in Design Pattern	Actual Name
Aggregate	Iterable
ConcreteAggregate	Inventory
Iterator	Iterator<Product>
ConcreteIterator	An iterator obtained from the ArrayList instance held by Inventory.
createIterator()	iterator()
next()	next()
isDone()	Opposite of hasNext()
currentItem()	Return value of next()

Observer Pattern



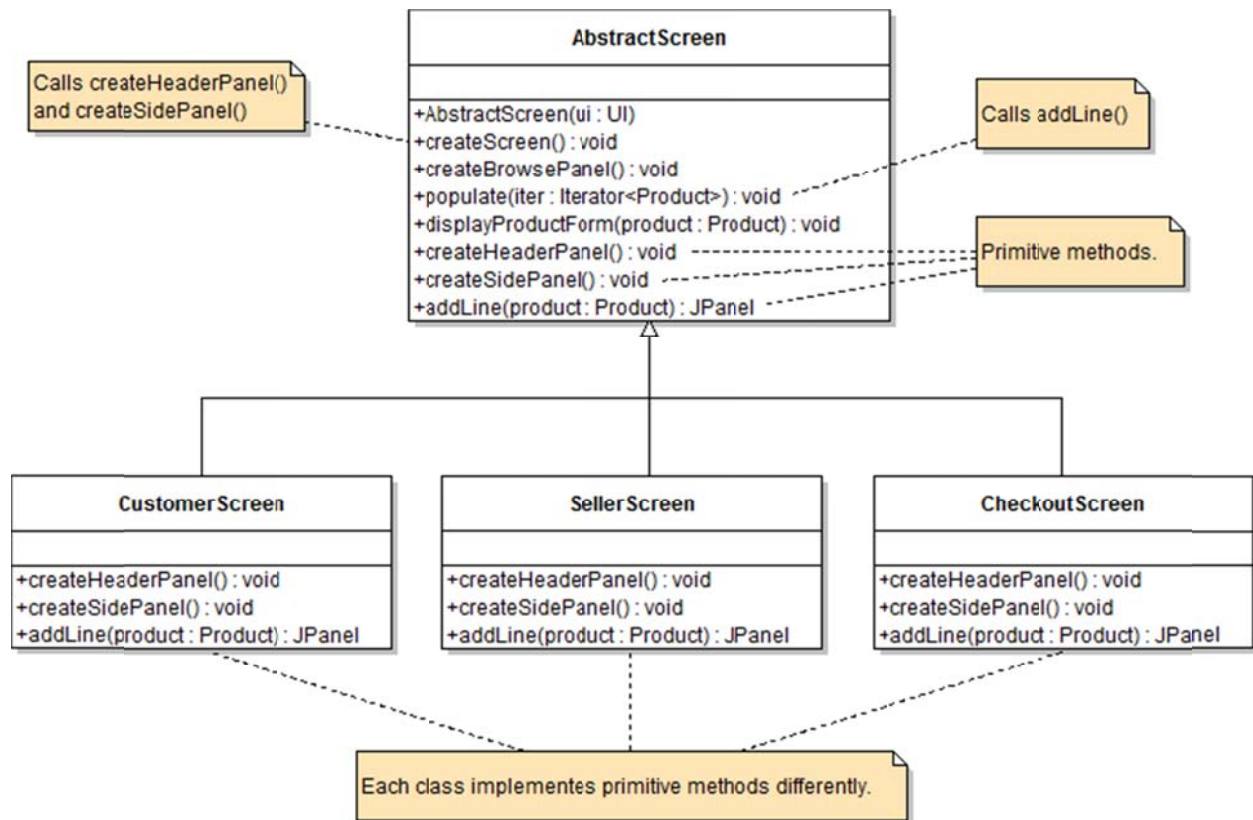
Name in Design Pattern	Actual Name
Subject	Product
Observer	ChangeListener
ConcreteObserver	An anonymous class that will call the <code>repaint()</code> method of, for example, a Product line JPanel.
<code>attach()</code>	<code>addListener()</code>
<code>notify()</code>	<code>stateChanged()</code>

Decorator Pattern



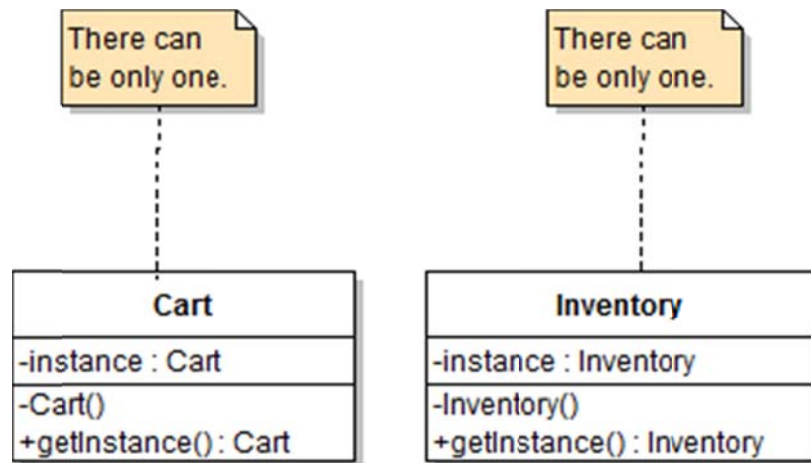
Name in Design Pattern	Actual Name
Component	Iterator<Product>
ConcreteComponent	Iterator from Inventory's ArrayList instance.
Decorator	PrunningIterator
method()	hasNext() and next()

Template Method Pattern



Name in Design Pattern	Actual Name
AbstractClass	AbstractScreen
ConcreteClass	CustomerScreen, SellerScreen, CheckoutScreen
templateMethod()	createScreen() (also populate())
primitiveOp1(), primitiveOp2()	createHeaderPanel(), createSidePanel (also addLine())

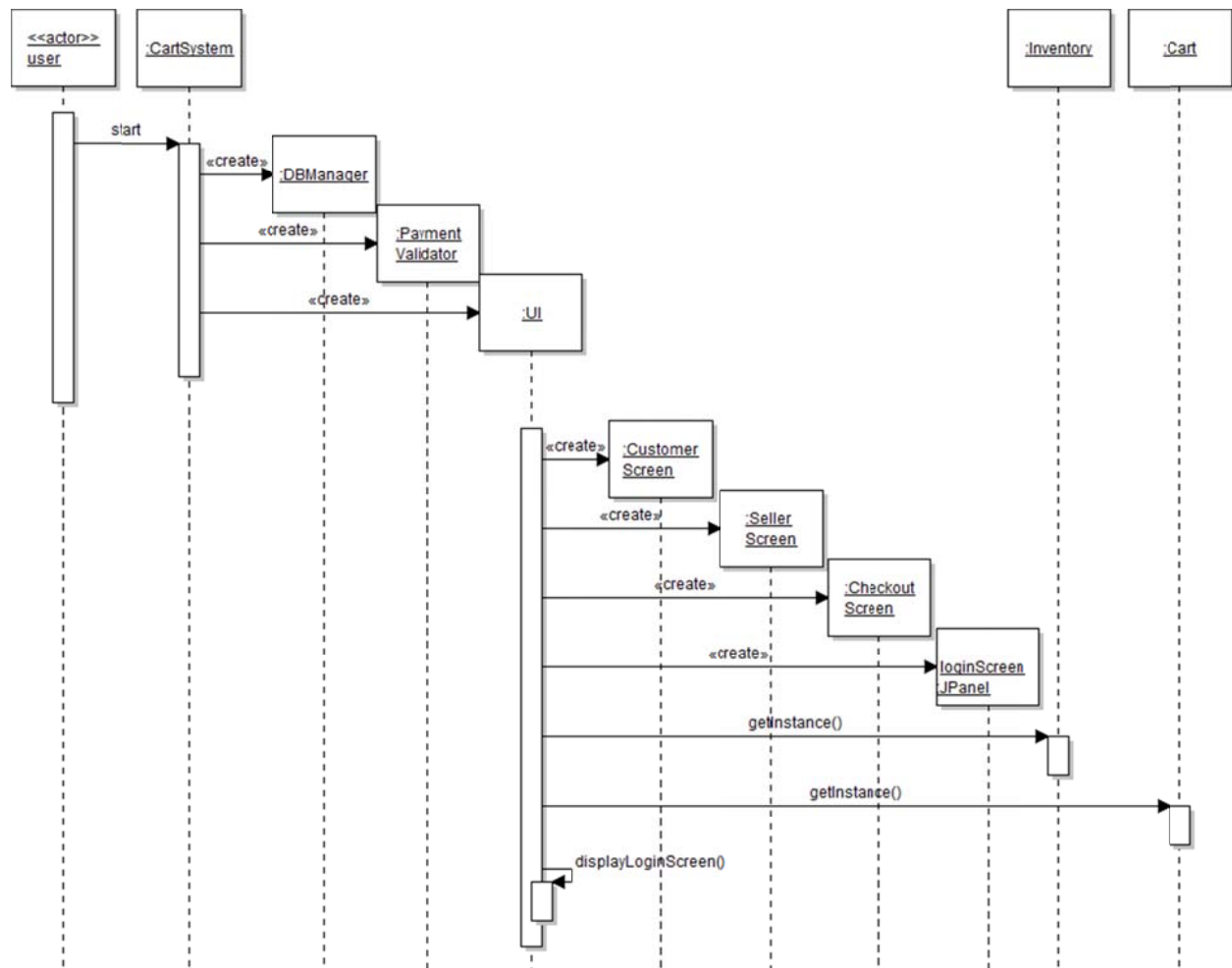
Singleton Pattern



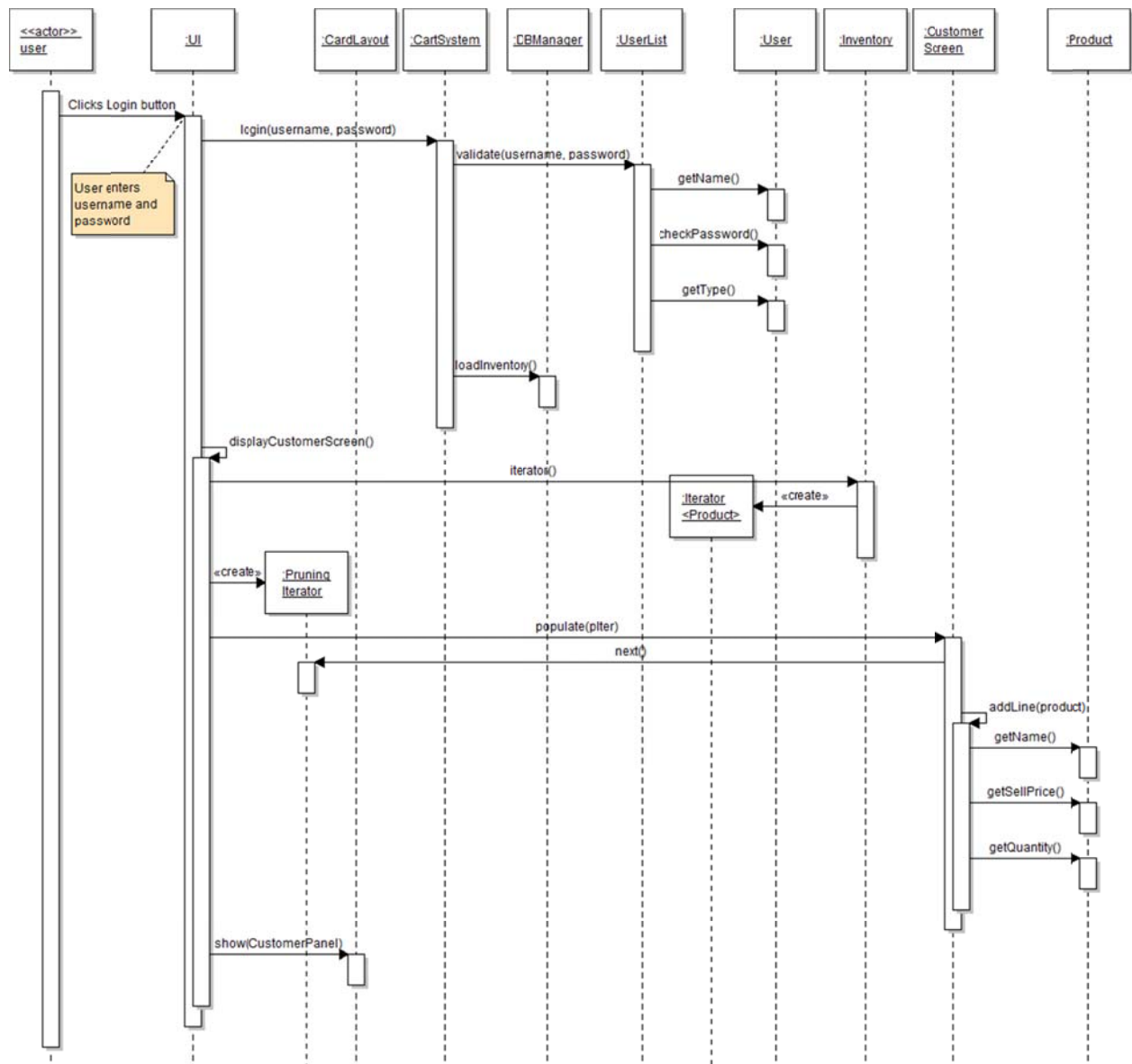
The Singleton Pattern guarantees that only one instance of **Cart** and **Inventory** classes can exist while the application runs.

Sequence Diagrams

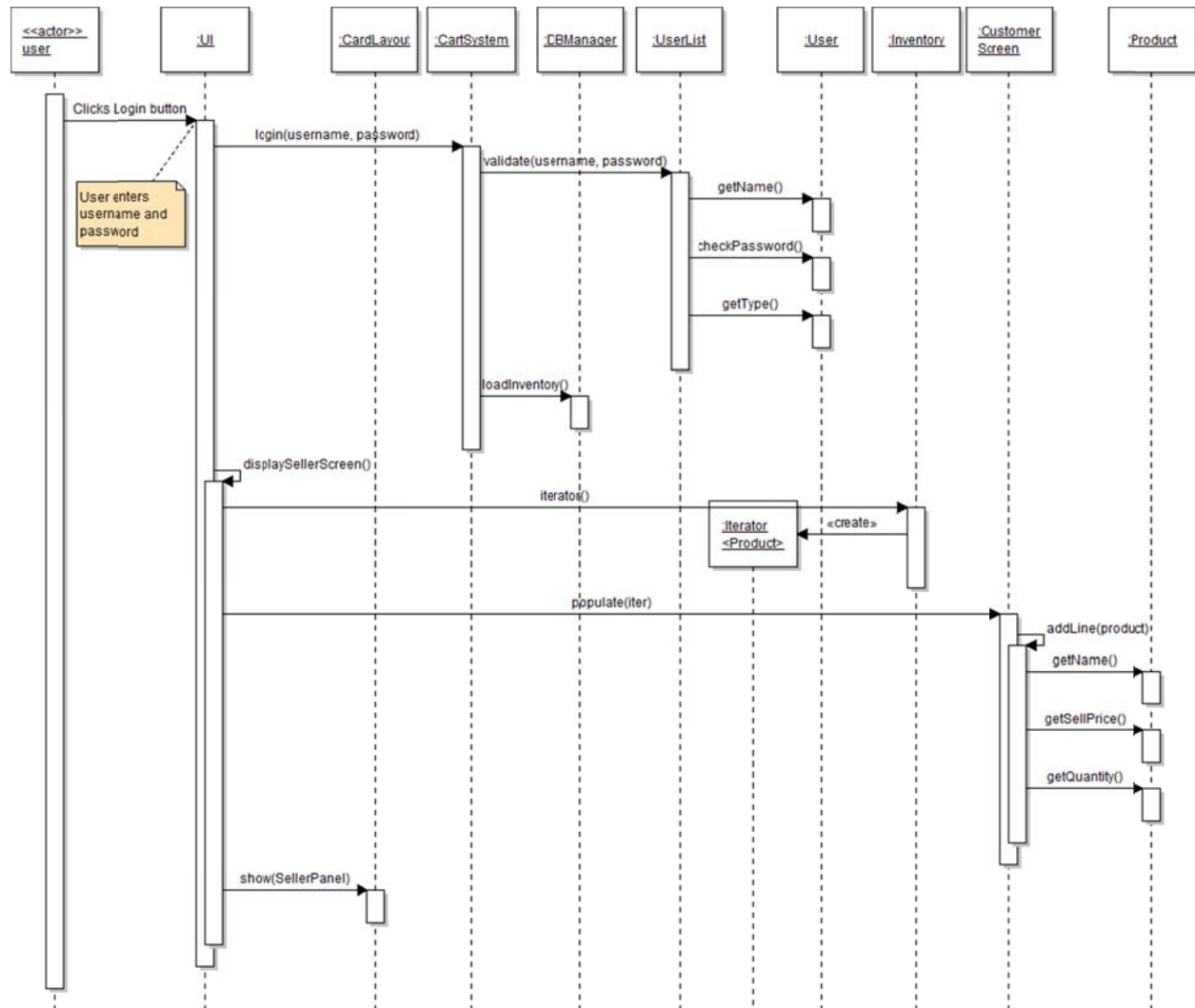
Startup



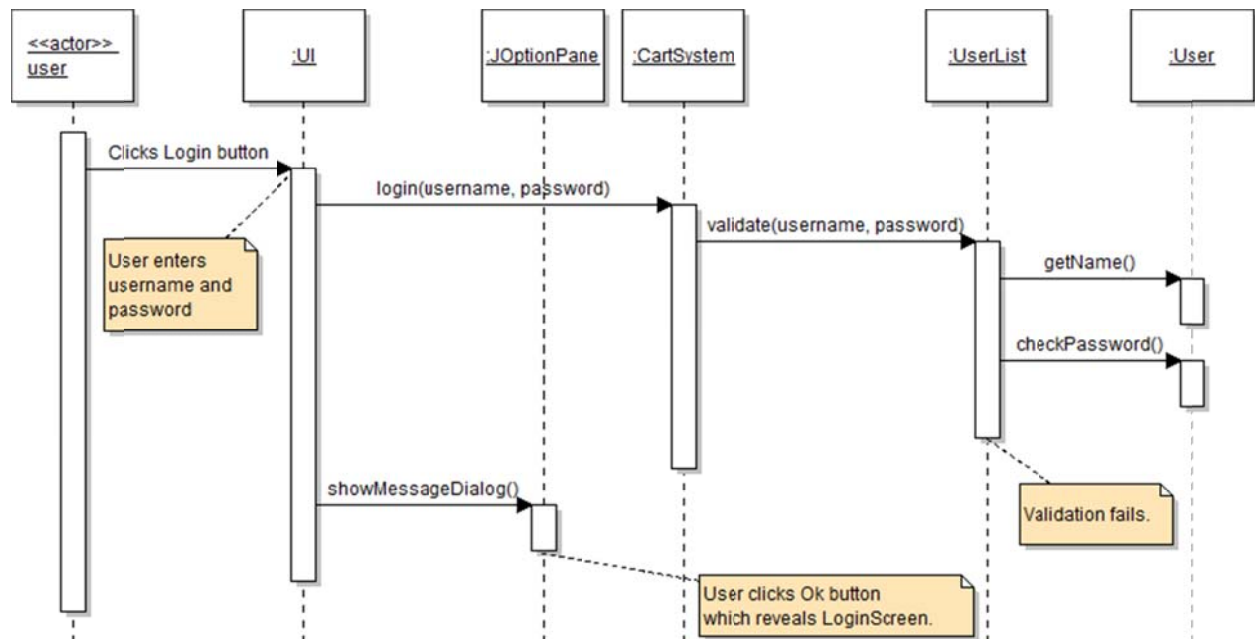
Log In



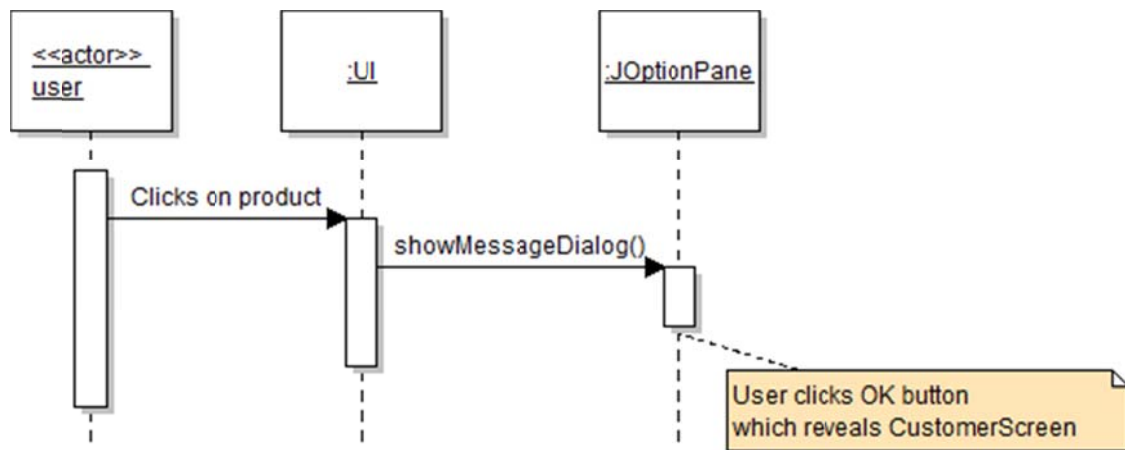
Variation #1. Login matches seller



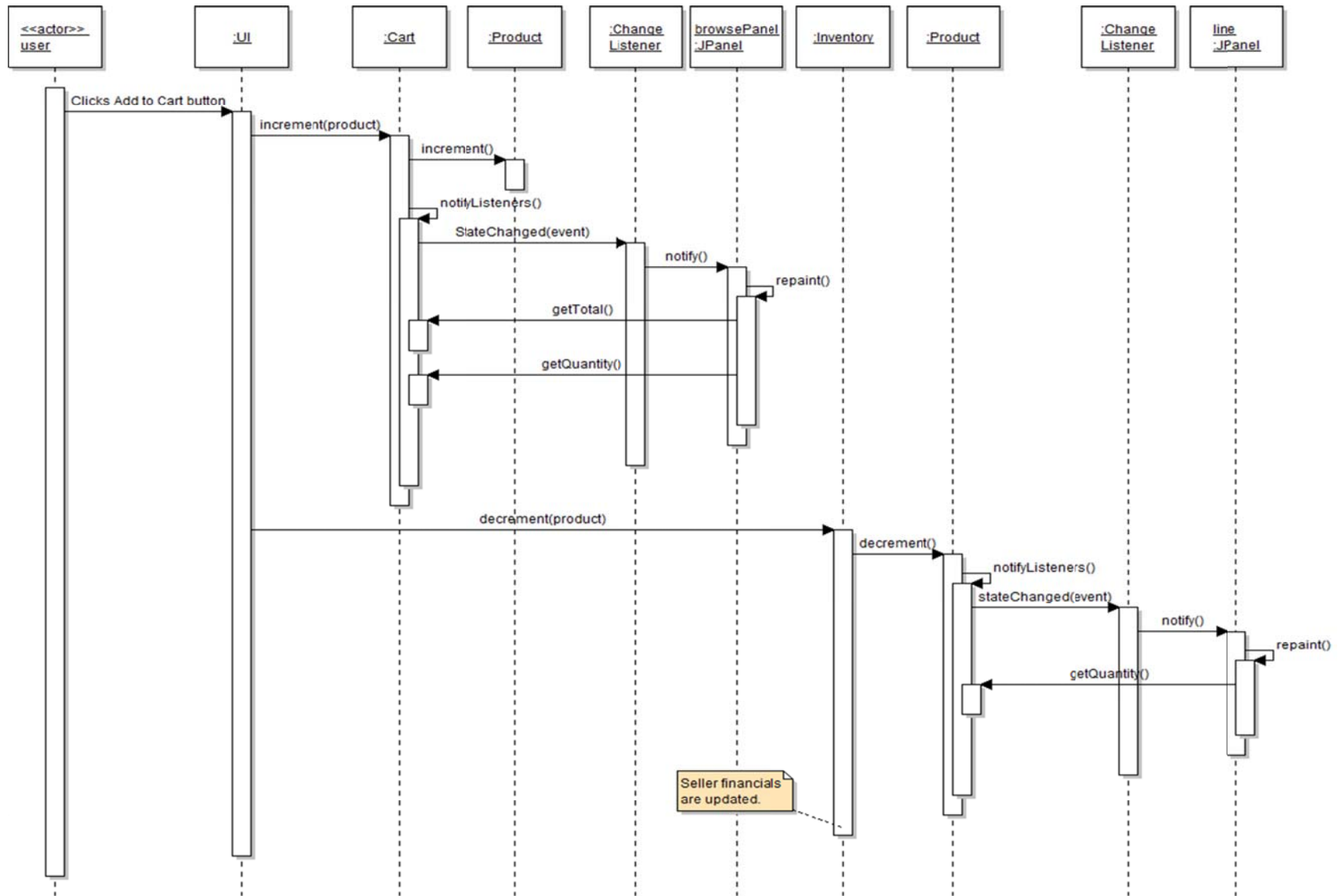
Variation #2. Validation fails.



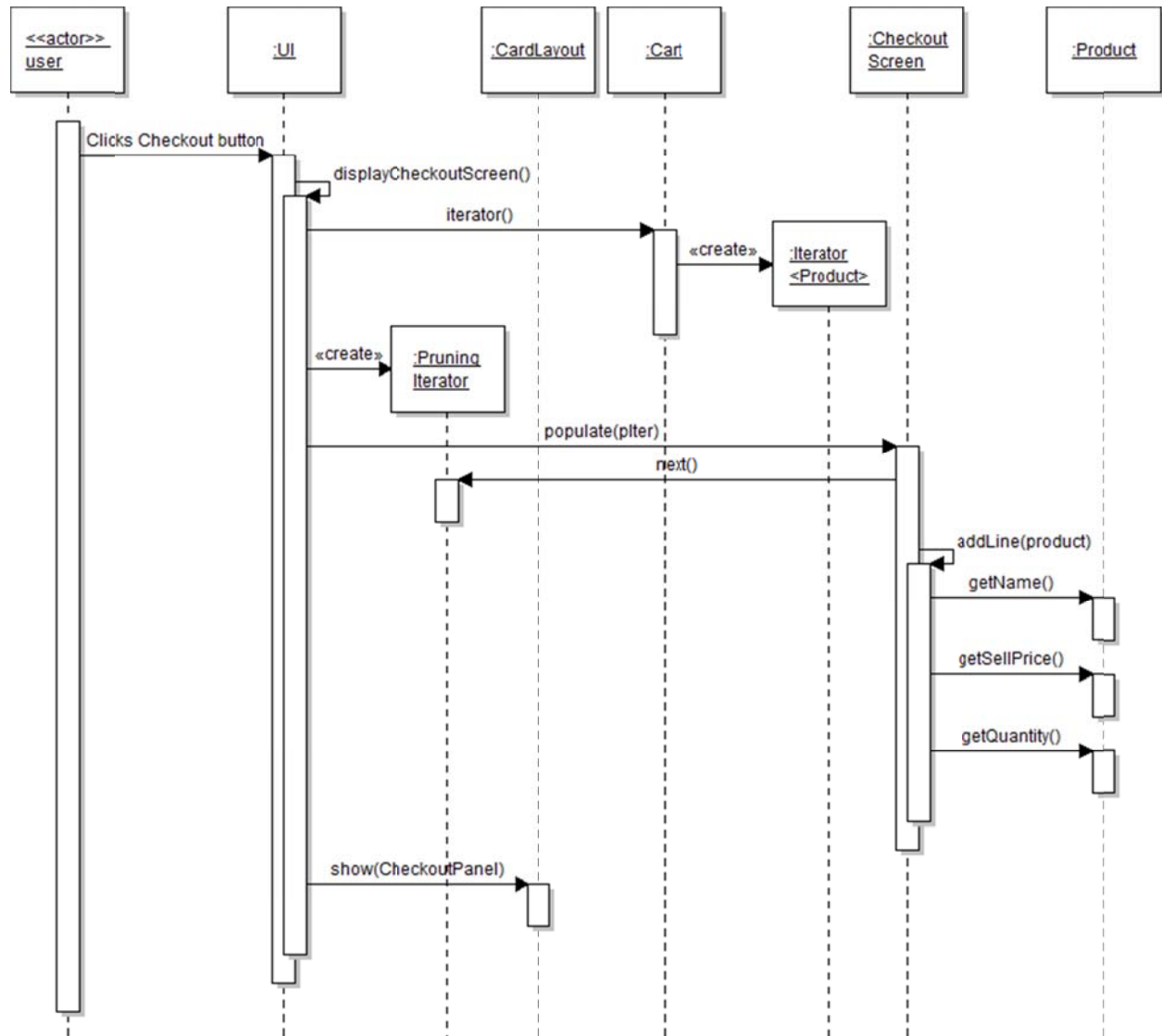
Review Product Details



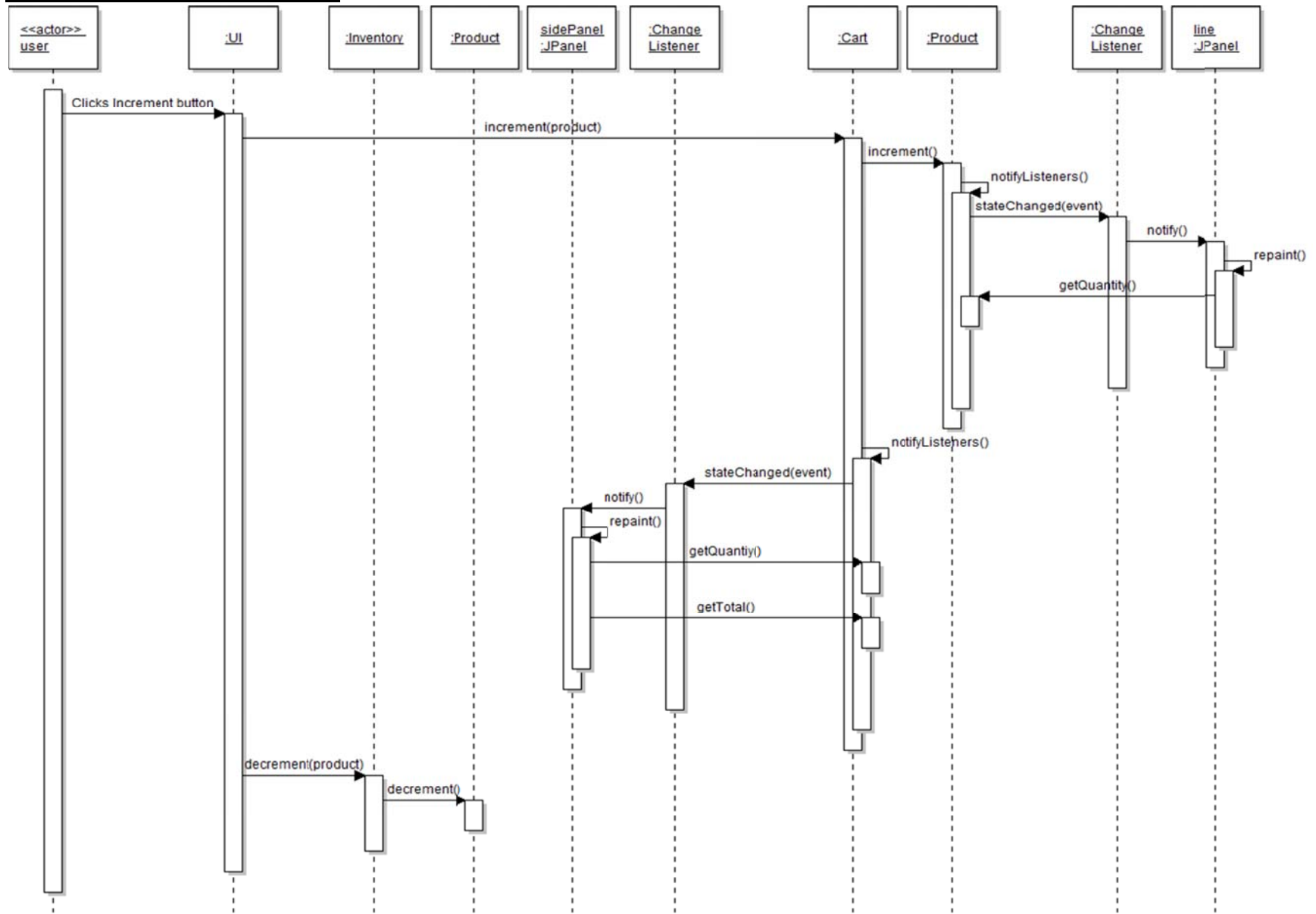
Add Product to Cart



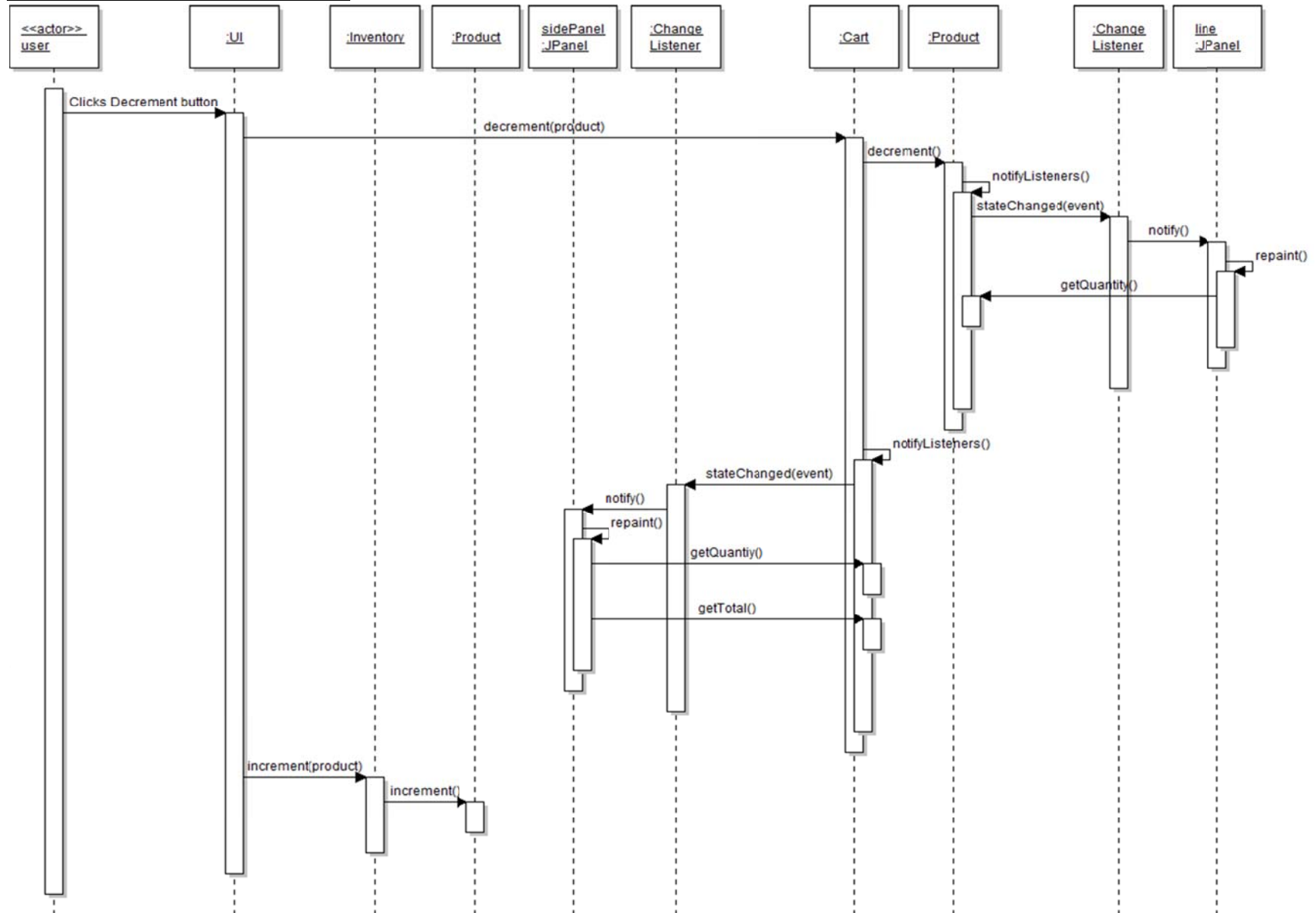
Checkout



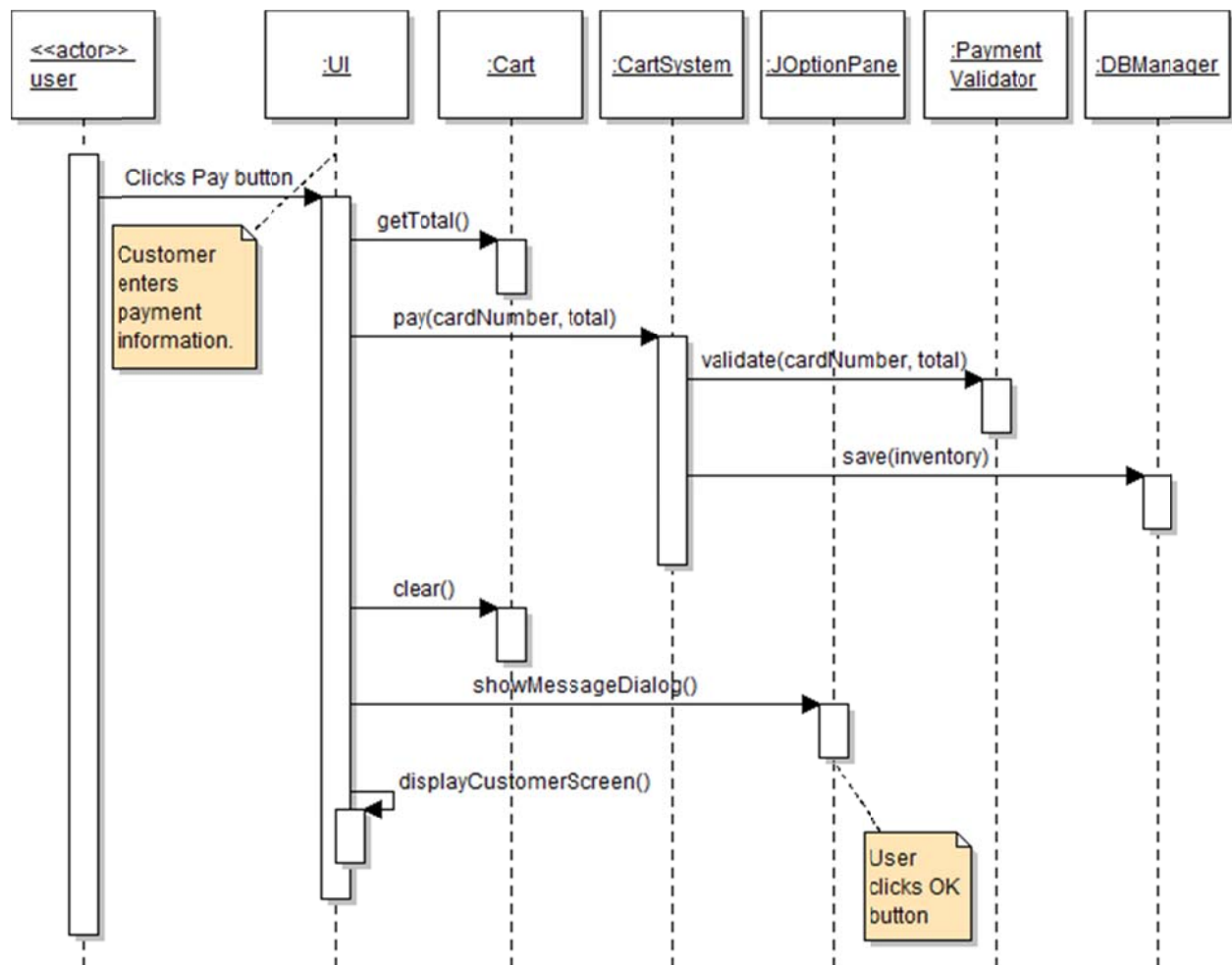
Variation #1. Increment Product



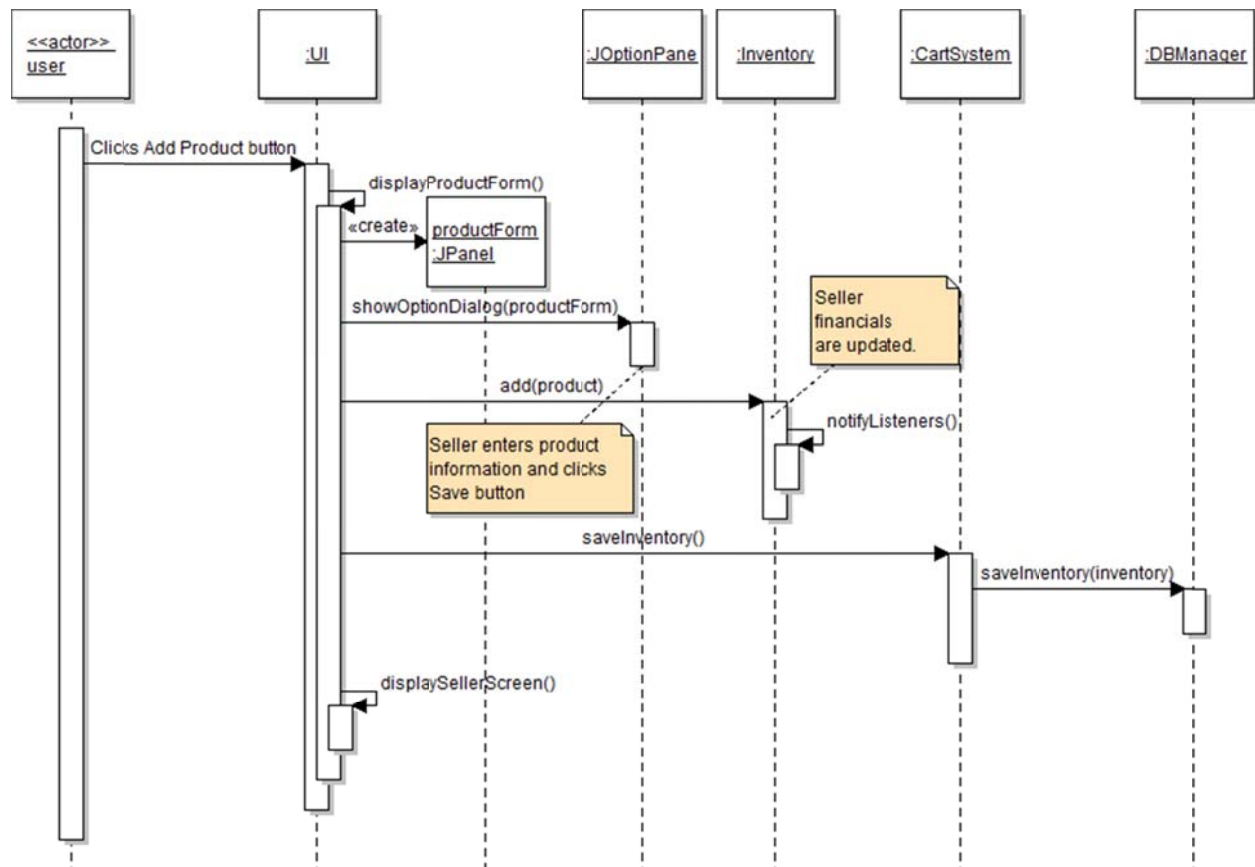
Variation #2. Decrement Product



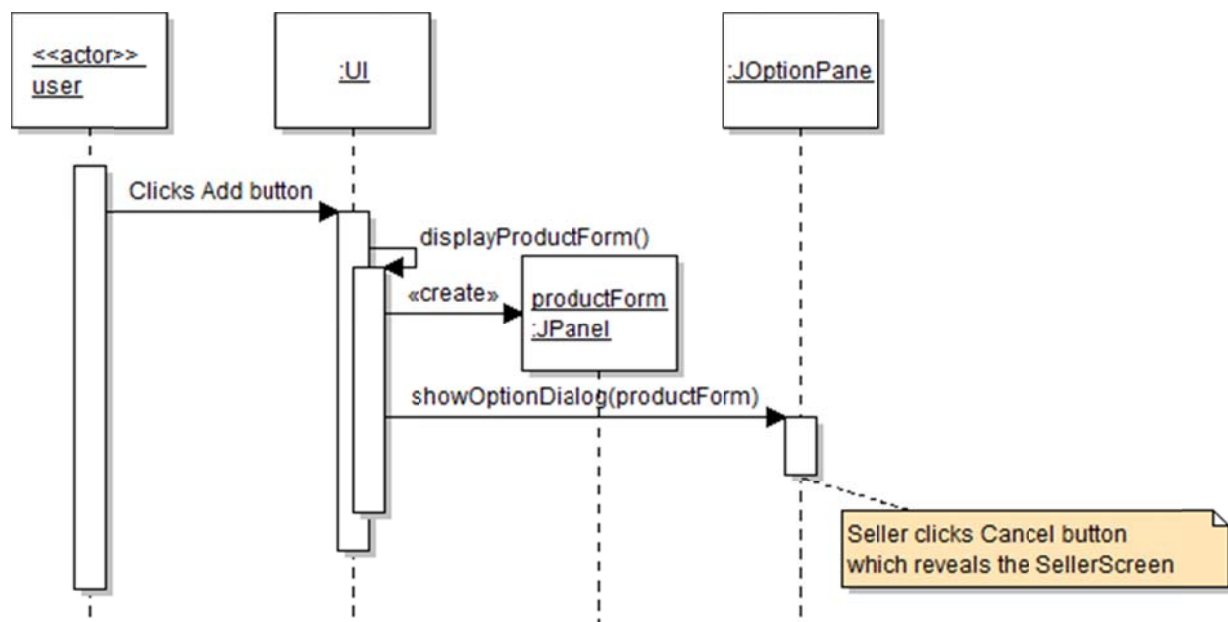
Pay



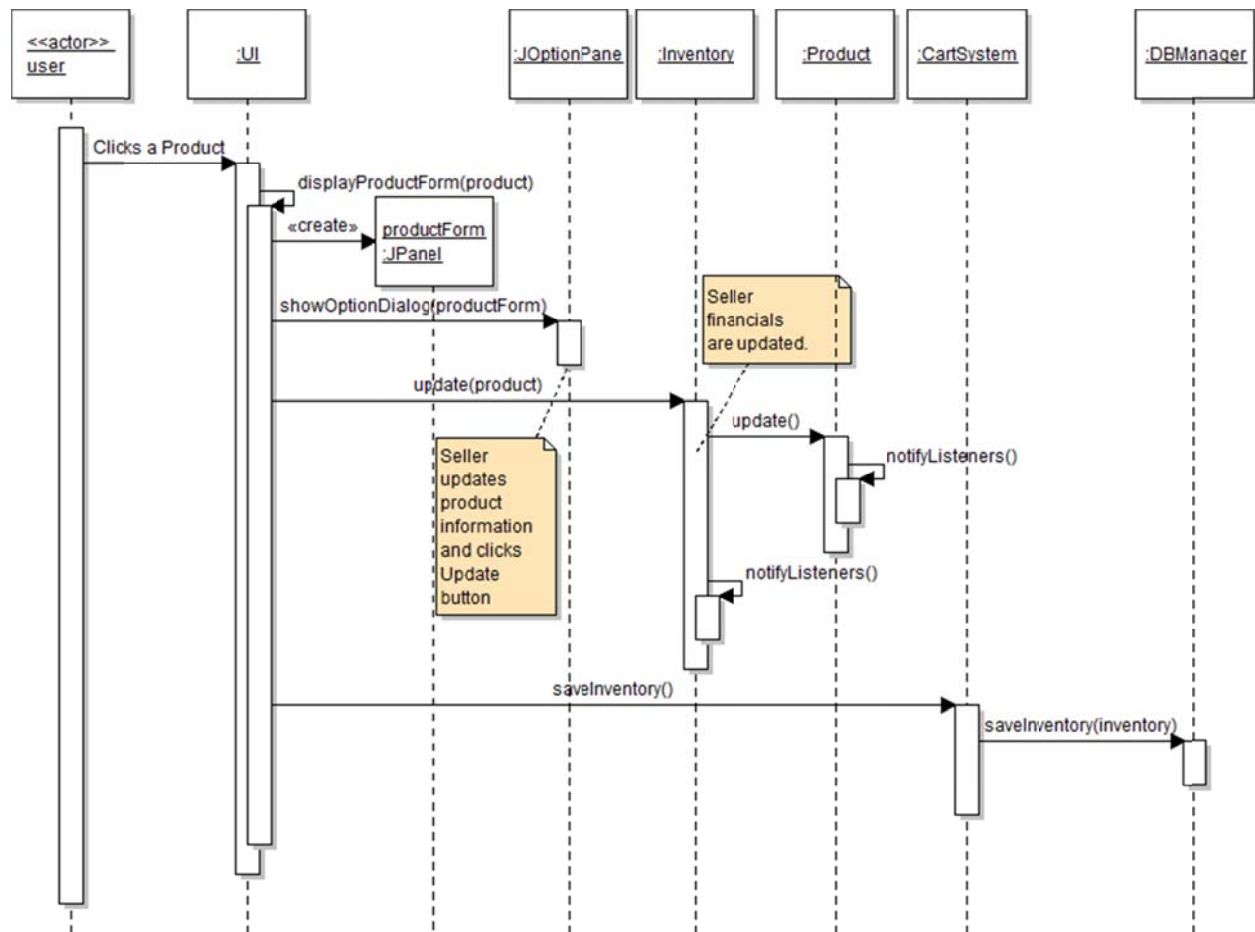
Add New Product



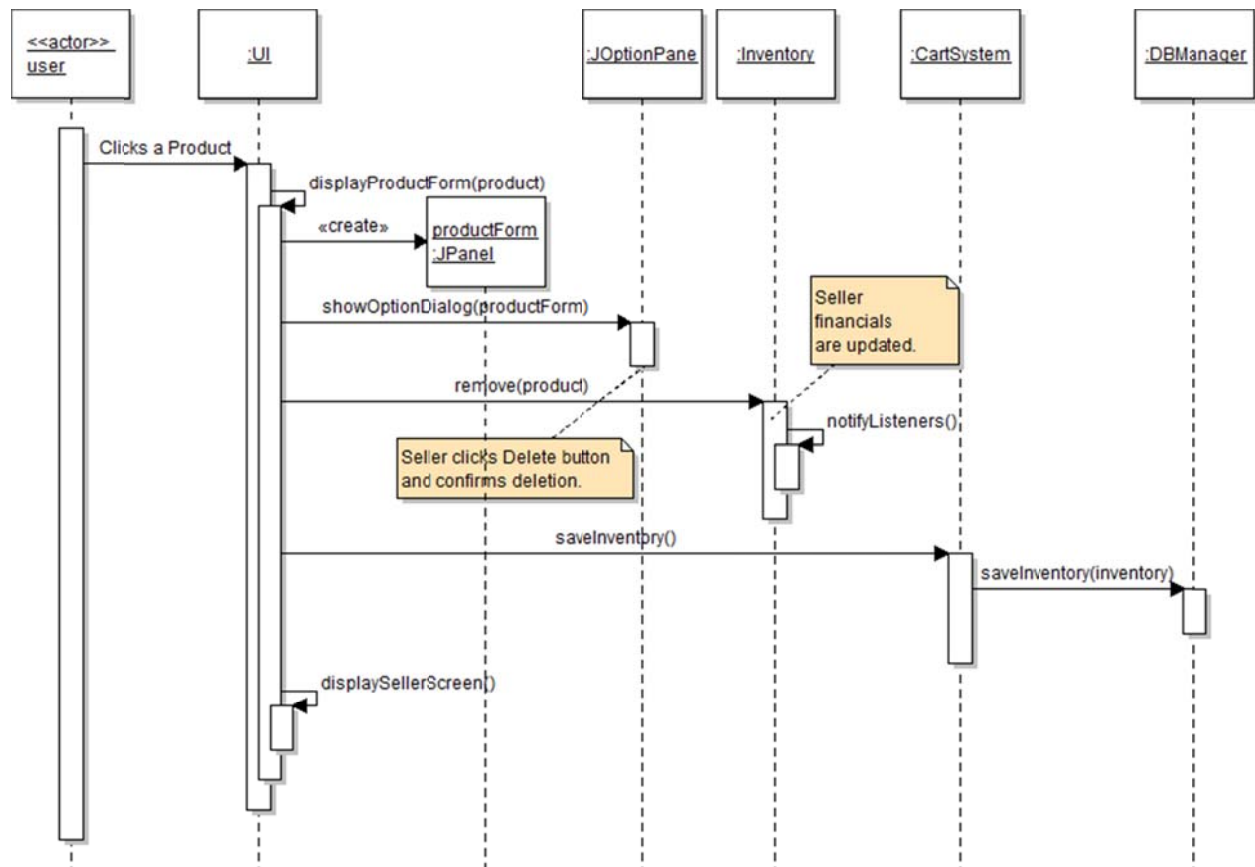
Variation #1. Cancel Add



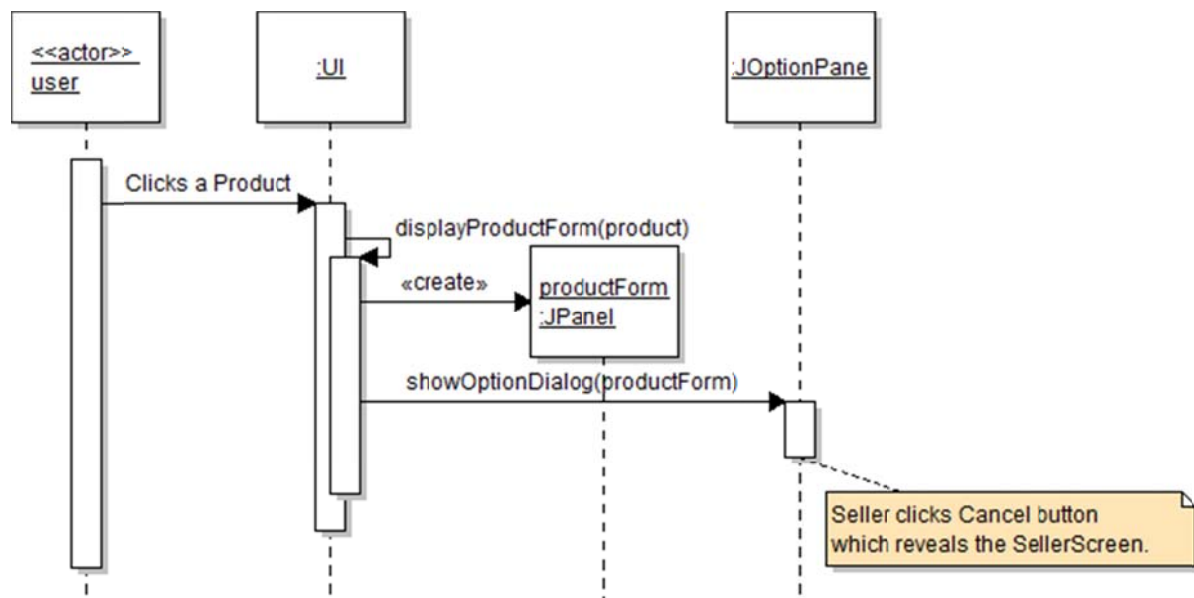
Update Product



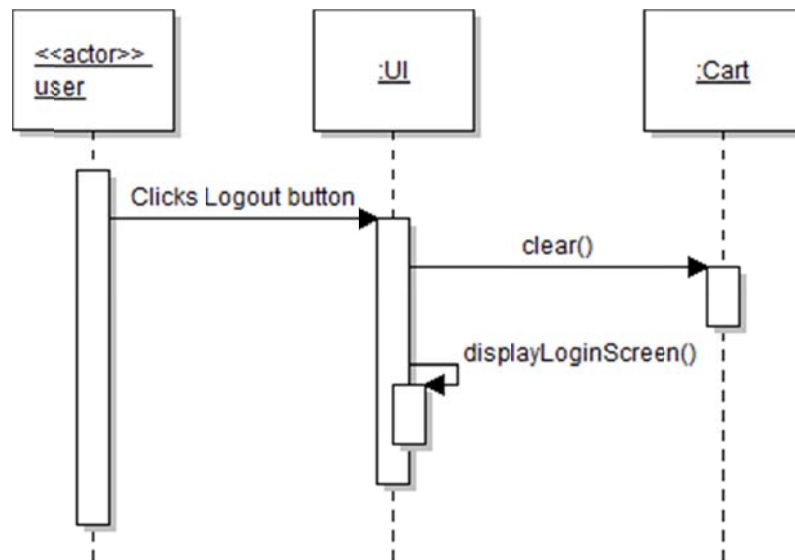
Variation #1. Delete Product



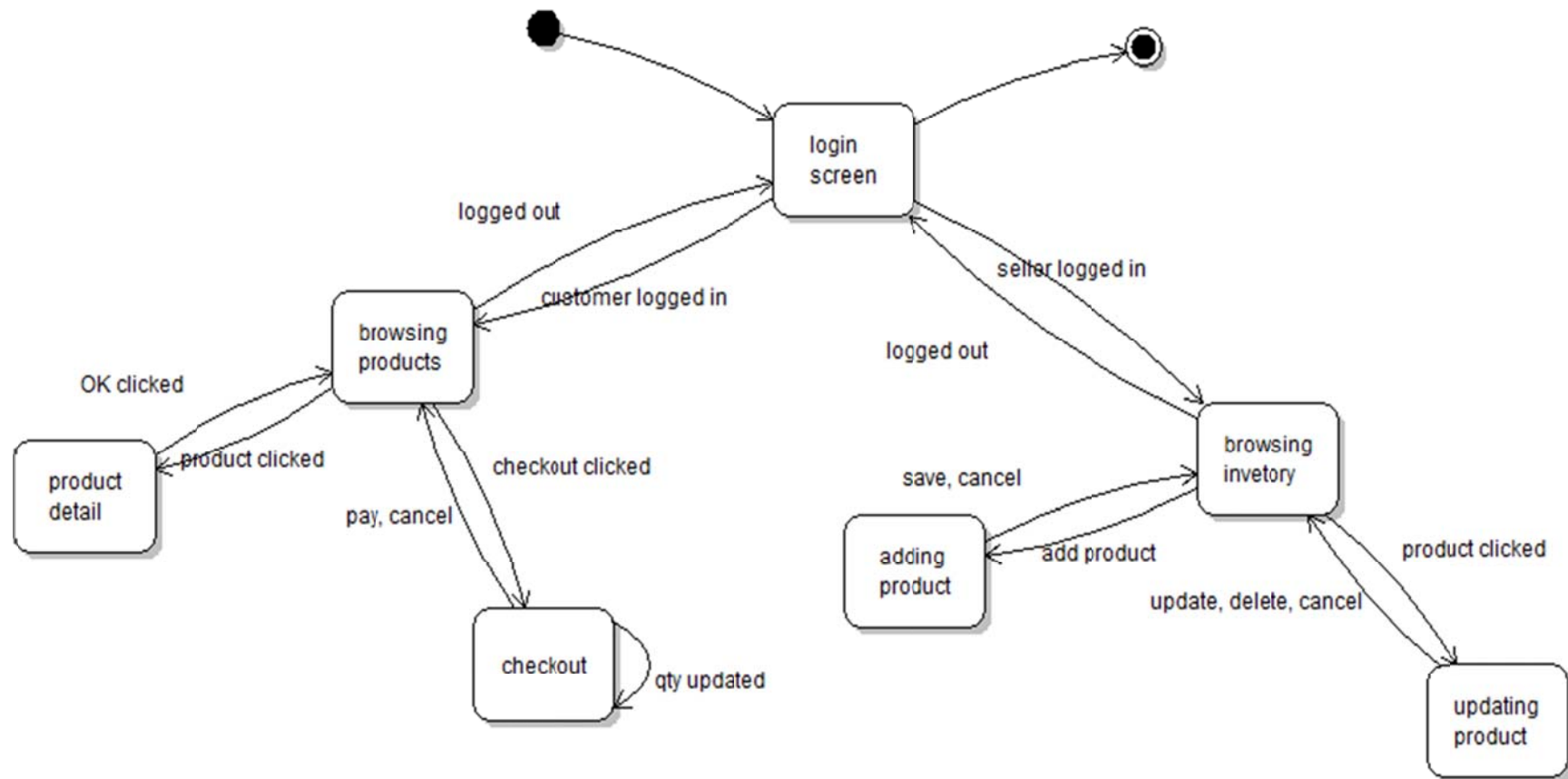
Variation #2. Cancel Update



Logout



State Diagram



Glossary

Cart:	The list of products that a Customer is preparing to buy.
Cart Summary:	The number of items and the total of Cart.
CartSystem:	The part of the Shopping Cart application that works behind the scenes.
Costs:	Sum of invoice price for all items added to the Seller's inventory.
Customer:	A user who is able to add items to the Cart, and purchase them.
DBManager:	The part of the Shopping Cart application that loads and saves the database.
Financials:	Seller's Costs, Revenues, and Profits.
Inventory:	A list of products and their available quantities.
PaymentValidator:	Third party application that validates payment information.
Product:	The representation of an item that includes ID, name, description, invoice price, sell price and quantity.
Profits:	Revenues – Costs.
Revenues:	Sum of sell price for all sold items.
Seller:	The user who has an inventory of items for sale.
UI:	User interface
User:	A person interacting with the Shopping Cart Application.