Title: SportStop (OOD---Group 15)

Team Members:

- Jesse Kelly
- Sadiki Brown
- Nick Bogdanovic

Shopping Cart User Interface Development Plan

Platform: Swift

Functional Specification:

- **Customer Functional Specification:**

After a customer logs in with a username and password, the user will be met with a window (frame) that opens where he/she can browse through a list of available products. Each product has a product name, price, and available quantity.

From this window the customer can select products and add them to the shopping cart, or they can click on a product and get the full product description, pricing and availability (quantity available) in another pop-up window. The customer can add the product to the shopping cart (quantity), depending on availability. The shopping cart total amount is kept current on the main product browse window.

The customer can proceed to checkout at any time. On the checkout window, the shopping cart can be updated by changing the item count for each product in the cart. At checkout the customer verifies the shopping cart content and pays for the goods by supplying the credit card. The application does not arrange for shipping.

- **Seller Functional Specification:**

When the seller logs in, a window opens where the current state of the inventory is shown. The seller can update the inventory by adding products - specifying product name, invoice price, sell price and by updating the available quantity.

The internal product representation includes ID, type, quantity, invoice price, and selling price. The application must keep track of all costs, revenues and profits.
The seller can access this information from the application UI.

Use Cases for Shopping Cart implementation:

1. User Logs In
2. Customer Adds Items to Shopping Cart
3. Customer Reviews Product Details
4. Customer Deletes from Shopping Cart
5. Customer Checks Out Successfully
6. Seller Reviews/Updates Inventory
7. Seller Adds New Product

1.

Use Case ID:	SC_UC_1
Use Case Name:	User Logs In
Created By:	Jesse Kelly
Description:	This use case allows the user to login to the system to access the relevant functions that adhere to the user's role. The two user roles are the customer and the seller. The customer and seller would need a username and password to login to the system and based on which role the user is logging into will determine what functionality will be available.
Primary Actor:	User
Secondary Actor:	None
Preconditions:	User already signed up with username and password
Postconditions:	User has access to the application functions

2.

Use Case ID:	SC_UC_2
Use Case Name:	Customer Add to Cart
Created By:	Jesse Kelly
Description:	Customer can add items to the shopping cart where a total price will be accumulated. After logging in, the window opens where the customer can browse a list of products with an option for selection that adds them to a shopping cart. The customer can adjust the quantity of items to be purchased.
Primary Actor:	Customer User
Secondary Actor:	None
Preconditions:	Customer has a successful login experience
Postconditions:	Customer has a shopping cart that is not empty

3.

Use Case ID:	SC_UC_3
Use Case Name:	Customer Review Product
Created By:	Jesse Kelly
Description:	The customer can click on a product and get the full product description, as well as the pricing and number of items available. Accessing the product description will be of use to the customers because it will help them decide whether the product will meet their criteria, or if they want to go back and continue browsing.
Primary Actor:	Customer User
Secondary Actor:	None
Preconditions:	The customer has clicked on a product
Postconditions:	The customer has access to the products attributes/information

4.

Use Case ID:	SC_UC_4
Use Case Name:	Customer Deletes from Shopping Cart
Created By:	Sadiki Brown
Description:	In this use case, the customer first logs in. Then, they browse to find the item they would like to purchase, and it's added to the cart. The customer decides they no longer want to purchase the item and go to their shopping cart, find the item, select “delete,” and is returned to the home screen.
Primary Actor:	Customer
Secondary Actor:	None
Preconditions:	An item has been added to the cart
Postconditions:	Cart has one less item than it did before

5.

Use Case ID:	SC_UC_5
Use Case Name:	Customer Checks Out Successfully
Created By:	Sadiki Brown
Description:	In this use case, the customer first logs in. Then, they browse to find the item they would like to purchase, and it's added to the cart. The customer decides to proceed to checkout and enter their payment information. An order confirmation is then displayed.
Primary Actor:	Customer
Secondary Actor:	None
Preconditions:	Customer wants to place an order
Postconditions:	Order has been placed

6.

Use Case ID:	SC_UC_6
Use Case Name:	Seller Reviews/Updates Inventory
Created By:	Nick Bogdanovic
Description:	Once logged in the seller is directed to a window with the current inventory of products and their quantity. The seller will go to any specific product and update its quantity and/or sell price.
Primary Actor:	Seller
Secondary Actor:	None
Preconditions:	Seller wants to review/update inventory
Postconditions:	Inventory has been reviewed and/or updated

7.

Use Case ID:	SC_UC_7
Use Case Name:	Seller Adds New Product
Created By:	Nick Bogdanovic
Description:	Once logged in the seller is directed to a window with the current inventory of products and their quantity. The seller will update this inventory by adding a product where they will be prompted to input product name, invoice price, selling price, and its quantity. Once done the product will be added
Primary Actor:	Seller
Secondary Actor:	None
Preconditions:	Seller wants to add a new product
Postconditions:	New product has been added

Glossary:

Application - Often used simply as a synonym for program. However, in Java, the term is particularly used of programs with a Graphical User Interface (GUI).

Application Programming Interface (API) - A set of definitions that you can make use of in writing programs. In the context of Java, these are the packages, classes, and interfaces that can be used to build complex applications without having to create everything from scratch.

Attribute - A particular usage of an instance variable. The set of attribute values held in a particular instance of a class define the current state of that instance. A class definition may impose constraints on the valid states of its instances by requiring that a particular attribute, or set of attributes, do not take on values. Attributes should be manipulated by accessor and mutator methods.

Class - A programming language concept that allows data and methods to be grouped together. The class concept is fundamental to the notion of an object-oriented programming language. The methods of a class define the set of permitted operations on the class's data (its attributes). This close tie between data and operations means that an instance of a class - an object - is responsible for responding to messages received via its defining class's methods.

Use Case - In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal. The actor can be a human or other external system.

Revised Functional Specification with GUI

A log-in frame is displayed containing two JTextAreas for the user's username and password and a JButton object labeled "log-in."

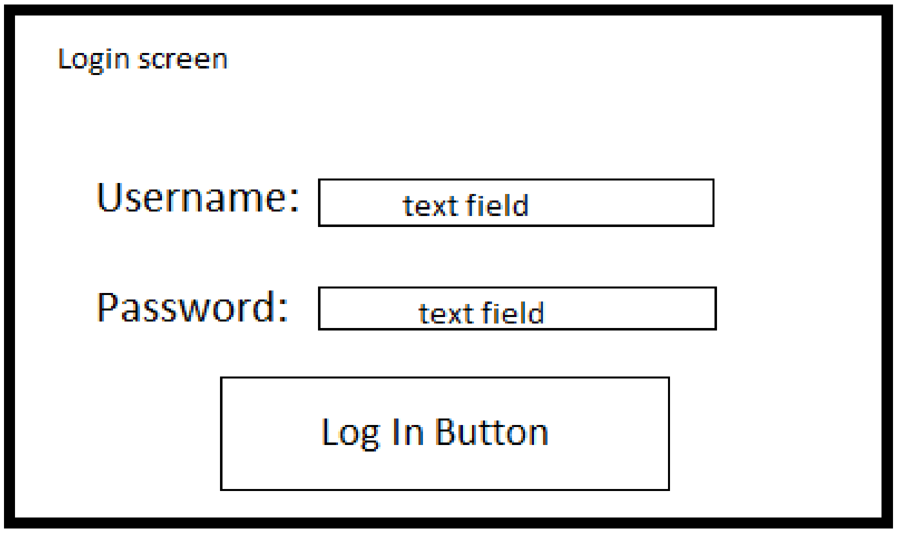
When the customer enters their username and password into the appropriate text fields and clicks "log in," they are met with a window (JFrame) that opens, displaying a list of available products (each with qualities - name, price, and quantity available) that also have two JButton objects. One labeled "Add to cart" and another labeled "View product info." At the bottom of the frame there is another JButton object labeled "Proceed to checkout." The customer can select "add to cart" (if quantity available is at least 1) or can select "Info" to access more information such as the full product description or the product qualities in another pop-up window. If the customer adds an item to the cart, the price of that item is added to the existing sale price (which starts at zero). This amount is kept current on the main window. At any point, the customer can choose "Proceed to checkout." In the checkout window, displaying a "cart" JFrame containing the list of products added to the cart by the customer. Each product contains a JTextArea labeled "Desired quantity" for if the customer wants to change the quantity from what it was. At the bottom of the frame there is a JButton labeled "Continue." At checkout, the customer will verify the cart contents by clicking "continue" which will open another JFrame called "Payment page" which displays JTextAreas for the card number, expiration date, and security code. At the bottom of the frame there is a JButton labeled "Order." The customer supplies their credit card information to appropriate text fields and presses "Order." To end the transaction, the customer receives a confirmation in another JFrame called "Confirmation page."

When the seller enters their username and password into the appropriate text fields and clicks "log in," they are met with a window (JFrame) displaying the list of products, each containing a JButton labeled "View Product Detains." Towards the top of the frame there are two JButtons labeled "Edit inventory" and "View financial info." The seller can update the inventory by clicking "Edit Inventory" which opens a JFrame called "Edit page" displaying the list of products currently in the inventory, each containing a JTextArea labeled “New quantity.” At the bottom, there are two JButtons labeled “Add new item” and “Update.” The seller can select “Add product” to add a completely new one which opens a JFrame called “Add page” displaying three JTextAreas to take in the new product’s name, invoice price, and sell price. At the bottom of the frame there is a JButton labeled “Update.” The internal product representation includes ID, type, quantity, invoice price, and selling price. The application will keep track of all costs, revenues, and profits which the seller can access from the application UI by clicking “Financial info” opening another JFrame.

<u>Use Cases:</u>

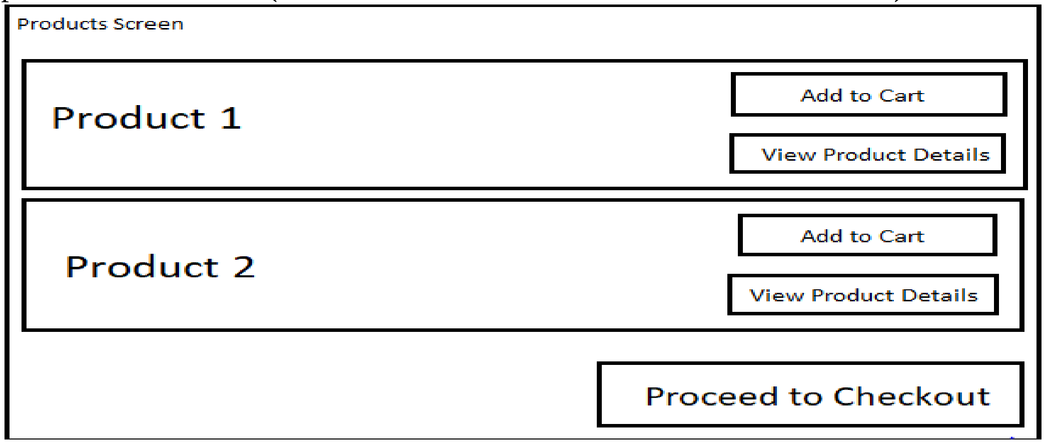
UC1: User Logs In:

1. User enters username and password in appropriate text fields and clicks “Log In”



UC2: Customer Adds Item to Shopping Cart:

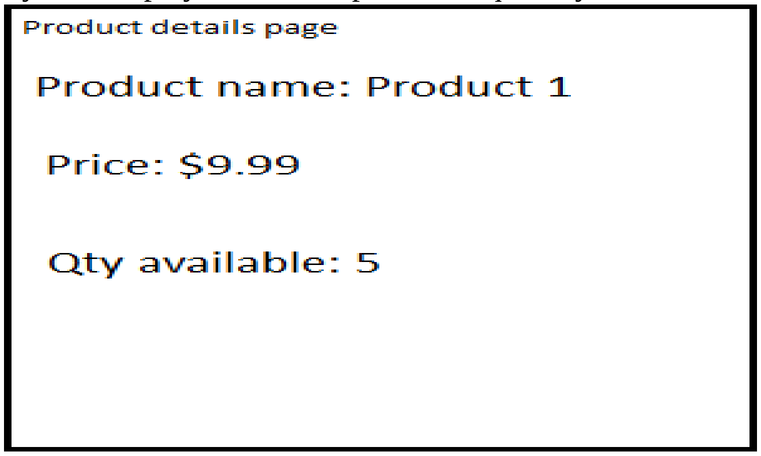
1. Customer carries out Log In
2. System displays the list of products with an option for selection that adds them to the shopping cart, an option to view product details, and an option at the very bottom to proceed to checkout (once at least one item has been added to the cart)



3. Customer selects "Add to cart" for Product 1
4. System adds Product 1 to cart

UC3: Customer Reviews Product Details:

1. Customer carries out Log In
2. System displays the list of products with an option for selection that adds them to the shopping cart and an option to view product details, and an option at the very bottom to proceed to checkout (once at least one item has been added to the cart)
   see above
3. Customer selects "View product details" for Product 1
4. System displays the name, price, and quantity available of the Product 1



Product details page

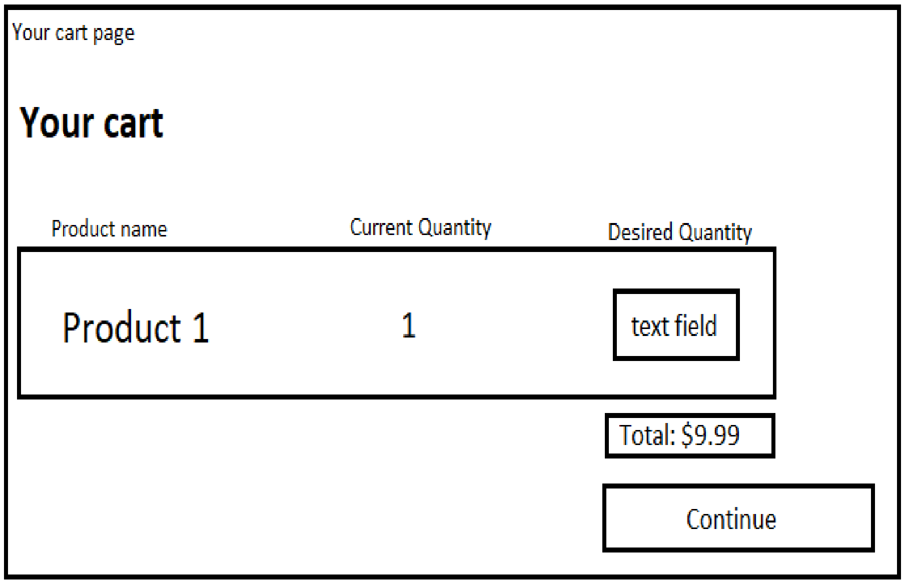
Product name: Product 1

Price: $9.99

Qty available: 5

UC4: Customer Checks Out Successfully:

1. Customer carries out Add Item to Shopping Cart and clicks "Proceed to Checkout"
2. System displays the current shopping cart, text fields for each product to edit the quantity if desired (given availability), and a continue button at the bottom



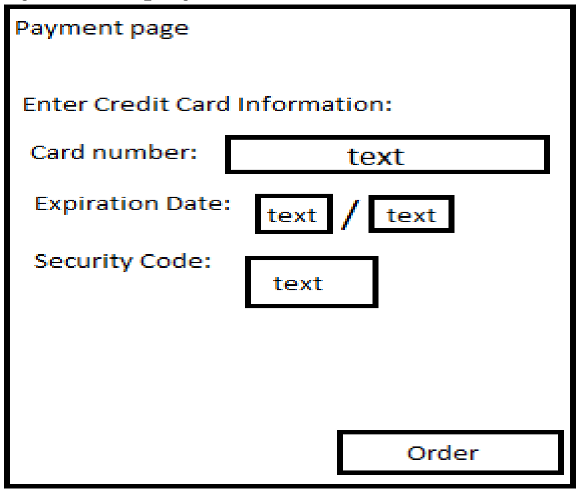
Your cart page

Your cart

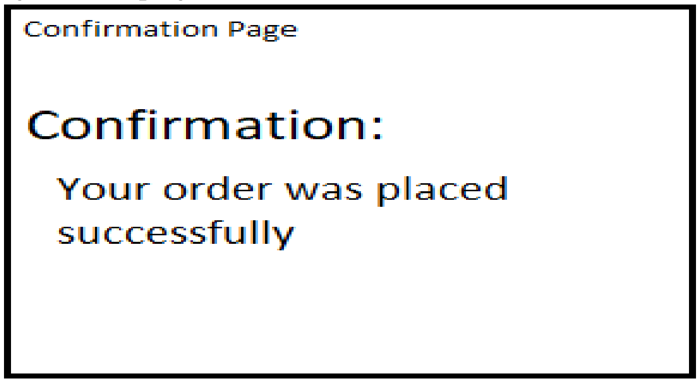
Product name	Current Quantity	Desired Quantity
Product 1	1	text field

Total: $9.99

3. Customer clicks “Continue”
4. System displays text fields for credit card information

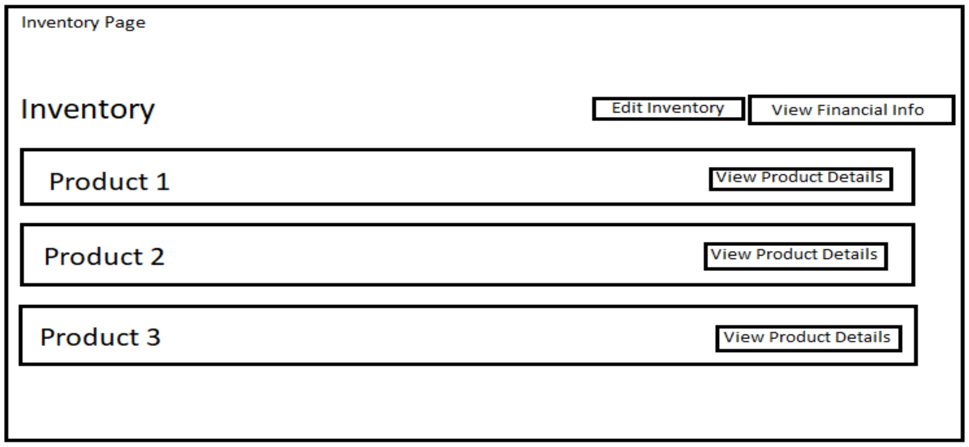


5. Customer enters credit card information and clicks “Order.”
6. System displays confirmation

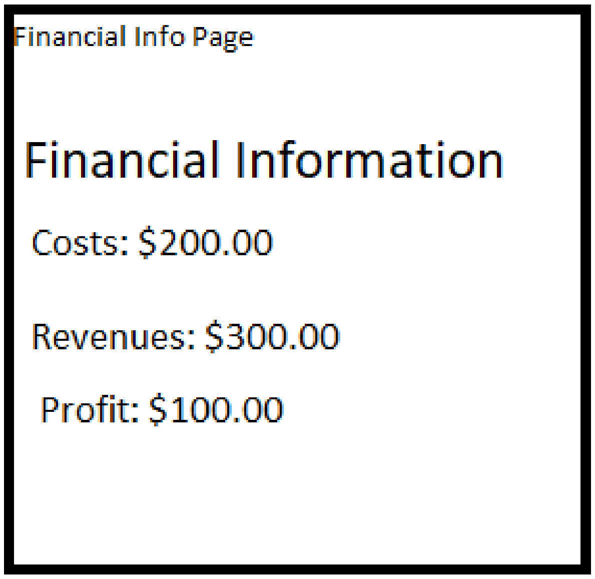


UC5: Seller Reviews Financial Information:

1. User carries out Log In
2. System displays the list of products in the seller’s inventory with an option for selection to edit the inventory, an option to view product details and an option to review financial information



3. Seller selects “View Financial Info”
4. System displays the seller’s costs, revenues, and profits



Financial Info Page

Financial Information

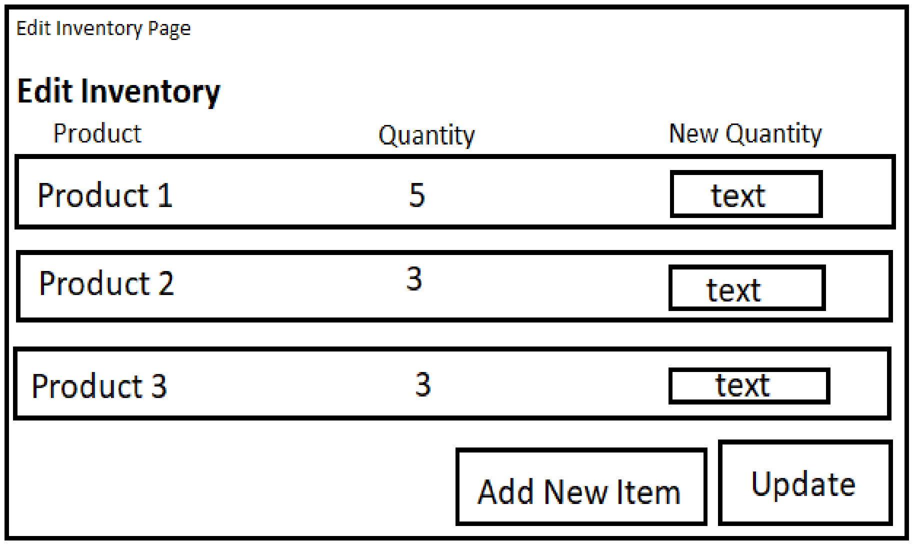
Costs: $200.00

Revenues: $300.00

Profit: $100.00

UC6: Seller Updates Inventory:

1. User carries out Log In
2. System displays the list of products in the seller’s inventory with an option for selection to edit the inventory, an option to view product details and an option to review financial information
   See above
3. Seller selects “Edit Inventory”
4. System displays current inventory with text fields to enter new quantities of existing items, a button to update those quantities, and a button to add a new item



Edit Inventory Page

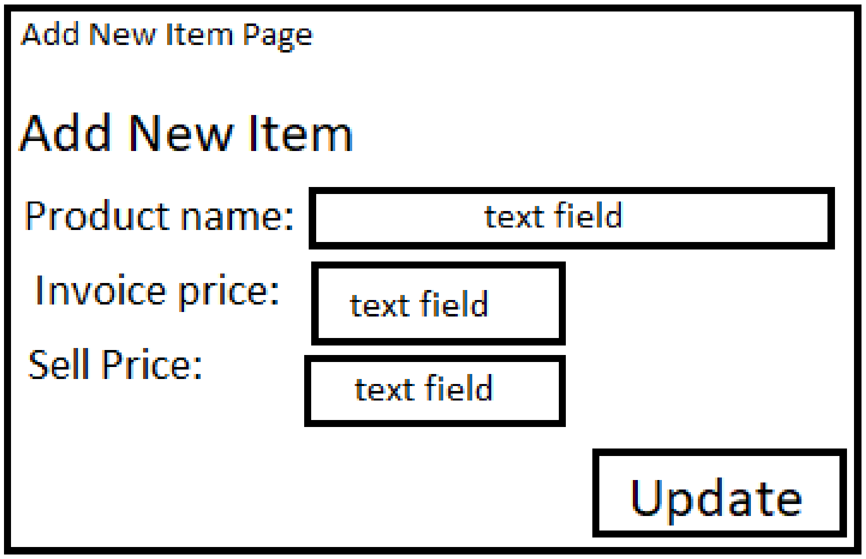
Edit Inventory

Product	Quantity	New Quantity
Product 1	5	text
Product 2	3	text
Product 3	3	text

5. Seller enters new quantity for an item and clicks “Update”
6. System updates inventory

UC7: Seller Adds New Product:

1. User carries out Log In
2. System displays the list of products in the seller's inventory with an option for selection to edit the inventory, an option to view product details and an option to review financial information
   see above
3. Seller selects "Edit Inventory"
4. System displays current inventory with text fields to enter new quantities of existing items, a button to update those quantities, and a button to add a new item
   see above
5. Seller selects "Add item"
6. System displays text fields for product information (product name, invoice price, sell price) and an update button to add the item to inventory and set its quantity to one



7. Seller enters product details and selects "Update"

<u>Use cases (Refined):</u>

UC1: User Logs In:

1. System displays JFrame(LoginPage)
2. User enters username into JTextArea(Username) and password into JTextArea(Password) then clicks JButton(LogIn)
3. ActionListener(forLogInButton) displays JFrame(ProductPage) or JFrame(InventoryPage)

UC2: Customer Adds Item to Shopping Cart:

1. Customer carries out Log In
2. ActionListener(forLogInButton) displays JFrame(ProductPage)
3. Customer clicks JButton(AddtoCart) for product 1

4. ActionListener(forAdddtoCartButton) adds product 1 to cart and updates total

UC3: Customer Reviews Product Details:

1. Customer carries out Log In
2. ActionListener(forLogInButton) displays JFrame(ProductPage)
3. Customer clicks JButton(ViewProductDetails) for product 1
4. ActionListener(forProductDetailsButton) displays JFrame(ProductDetailsPage)

UC4: Customer Checks Out Successfully:

1. Customer carries out Add Item to Shopping Cart and clicks JButton(ProceedtoCheckout)
2. ActionListener(forProceedtoCheckoutButton) displays JFrame(CartPage)
3. Customer clicks JButton(Continue)
4. ActionListener(forContinueButton) displays JFrame(PaymentPage)
5. Customer enters credit card number into JTextArea(CCN), expiration date month into JTextArea(ExpMonth) and year into JTextArea(ExpYear), and the security code into JTextArea(SecurityCode) then clicks JButton(Order)
6. ActionListener(forOrderButton) displays JFrame(ConfirmationPage)

UC5: Seller Reviews Financial Information:

1. Seller carries out Log In
2. ActionListener(forLogInButton) displays JFrame(InventoryPage)
3. Seller clicks JButton(ViewFinancialInfo)
4. ActionListener(forViewFinancialInfoButton) displays JFrame(FinancialInfoPage)

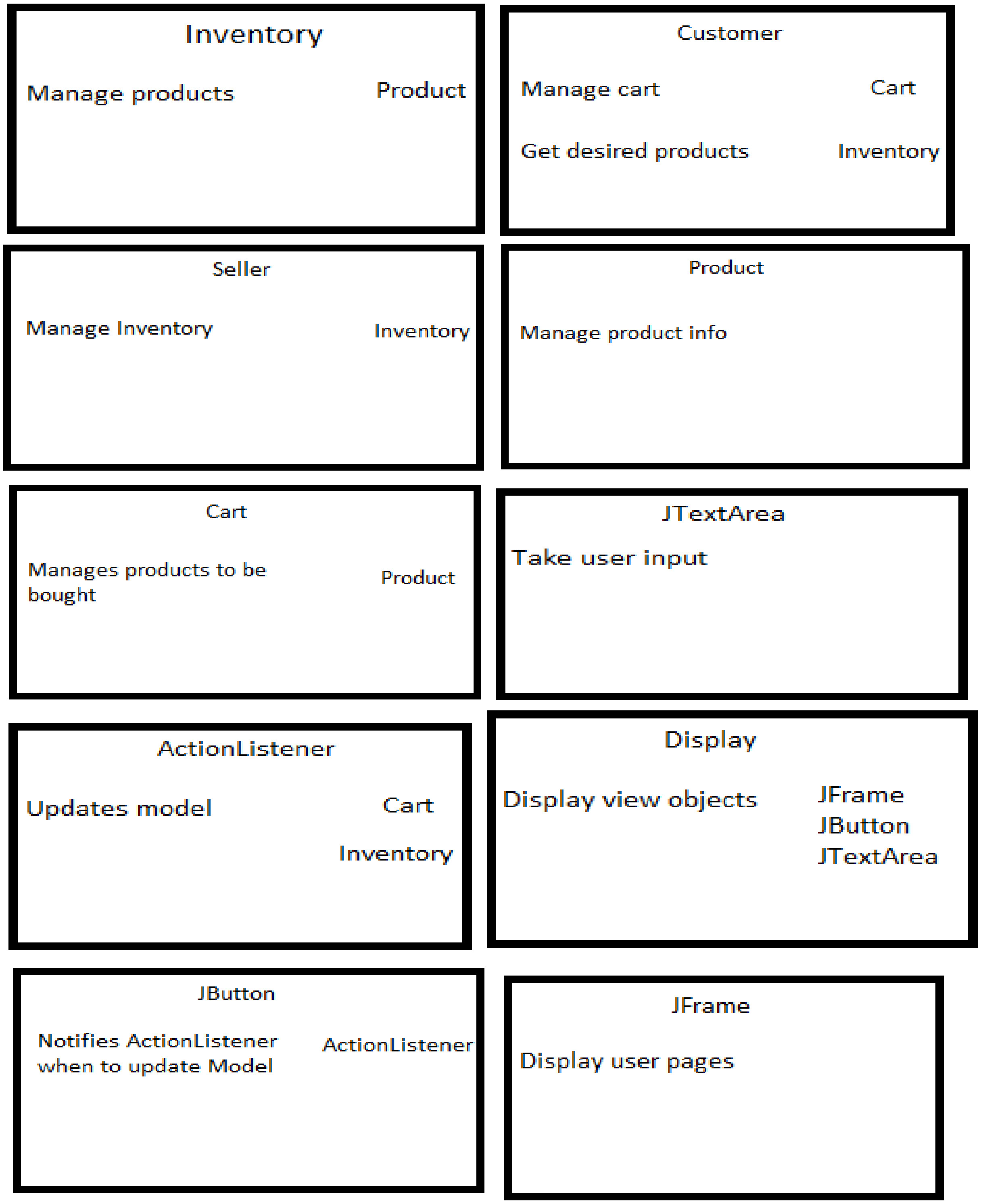
UC6: Seller Updates Inventory:

1. Seller carries out Log In
2. ActionListener(forLogInButton) displays JFrame(InventoryPage)
3. Seller clicks JButton(EditInventory)
4. ActionListener(forEditInventoryButton) displays JFrame(EditInventoryPage)
5. Seller enters new quantity in JTextArea(NewQuantity) for product 1 and clicks JButton(Update)
6. ActionListener(forUpdateButton) updates inventory

UC7: Seller Adds New Product:

1. Seller carries out Log In
2. ActionListener(forLogInButton) displays JFrame(InventoryPage)
3. Seller clicks JButton(EditInventory)
4. ActionListener(forEditInventoryButton) displays JFrame(EditInventoryPage)
5. Seller clicks JButton(AddItem)
6. ActionListener(forAddItemButton) displays JFrame(AddNewItemPage)
7. Seller enters new product's name into JTextArea(ProductName), invoice price into JTextArea(InvoicePrice), and sell price to JTextArea(SellPrice) then clicks JButton(Update)
8. ActionListener(forUpdateButton) updates Inventory with new item

CRC Cards:



Inventory

Manage products	Product

Customer

Manage cart	Cart
Get desired products	Inventory

Seller

Manage Inventory	Inventory

Product

Manage product info	

Cart

Manages products to be bought	Product

JTextArea

Take user input	

ActionListener

Updates model	Cart
	Inventory

Display

Display view objects	JFrame JButton JTextArea

JButton

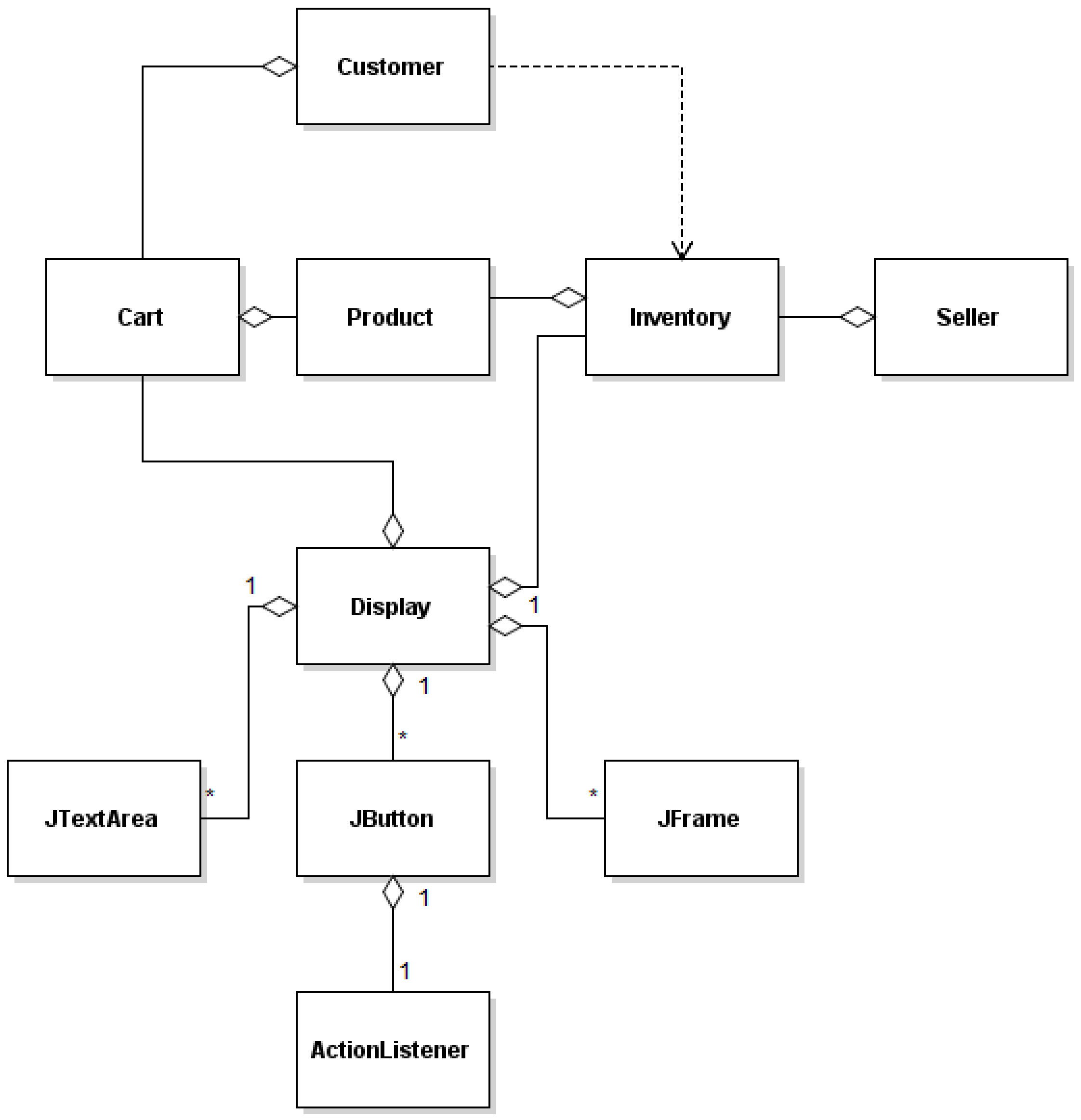
Notifies ActionListener when to update Model	ActionListener

JFrame

Display user pages	

Models – Cart, Inventory; Views – JFrame, JButton, JTextArea; Controller - ActionListener

Class Diagram:

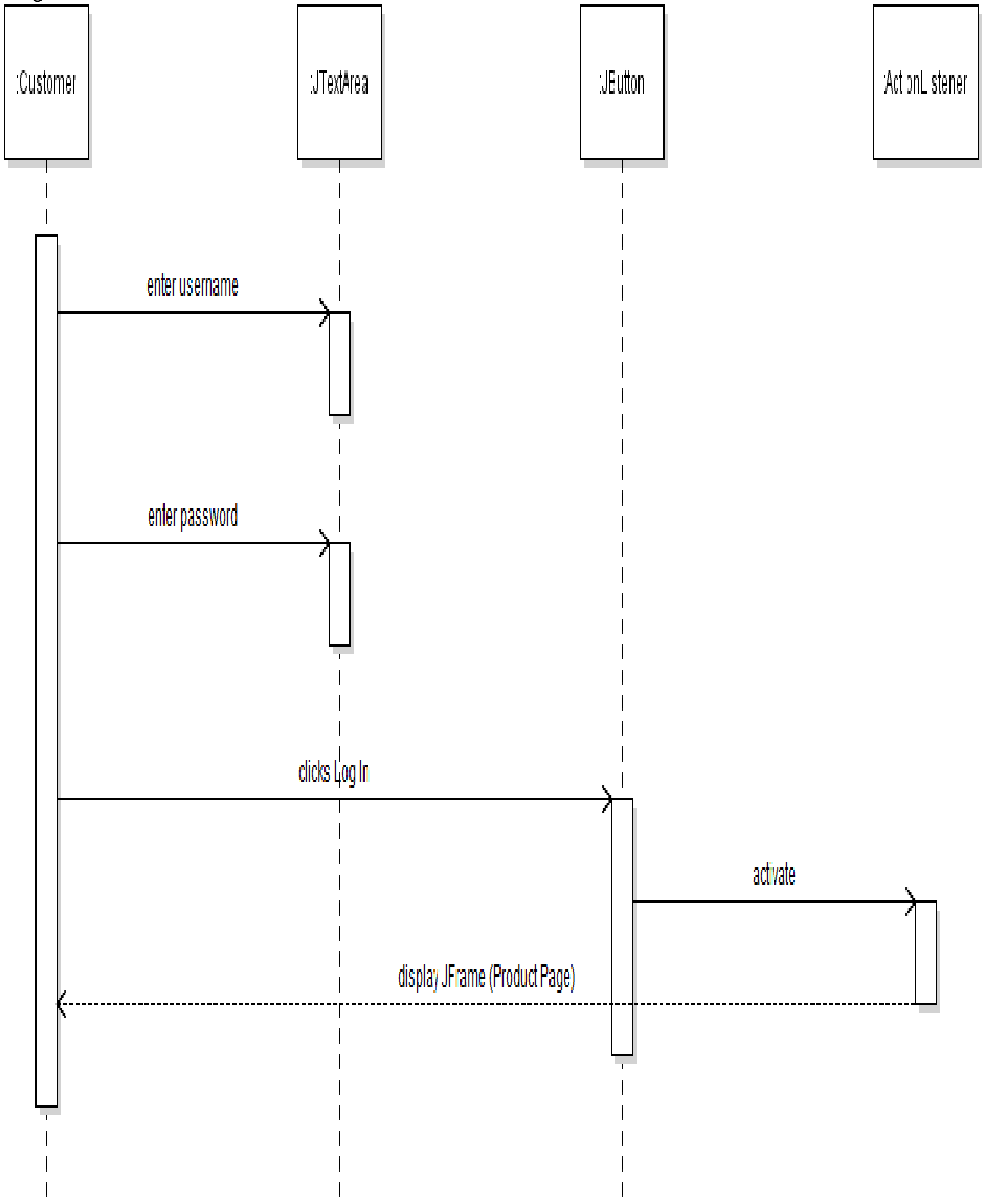


Patterns:

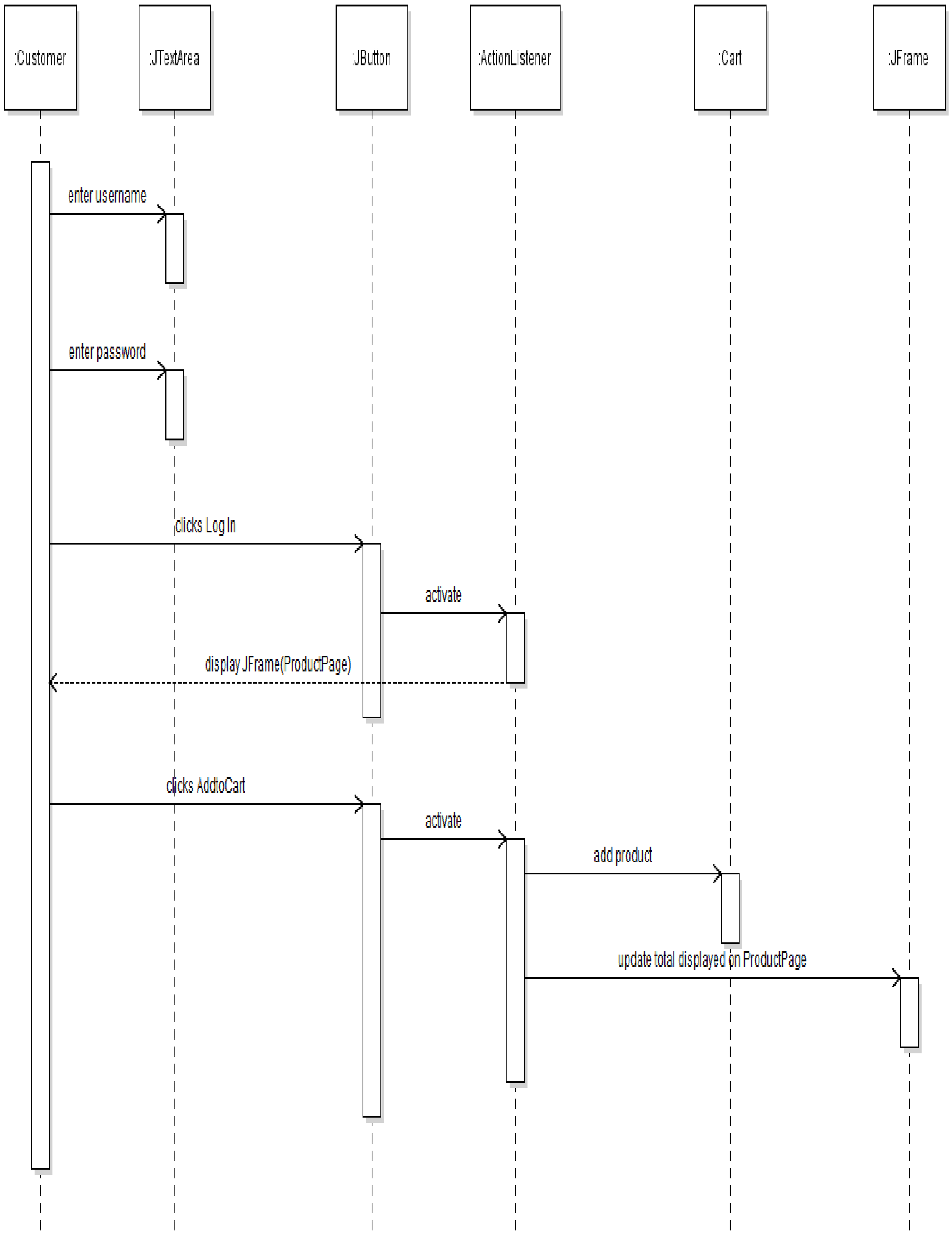
1. Observer – MVC for UI and notifying Model when object is added/removed
2. Iterator – Iterate through inventory (add costs)
3. Composite – Bundle together line items
4. Decorator – Handle discounted item just like any other item
5. Strategy – Different algorithms for seller/customer that implement abstract interface

Sequence Diagrams:

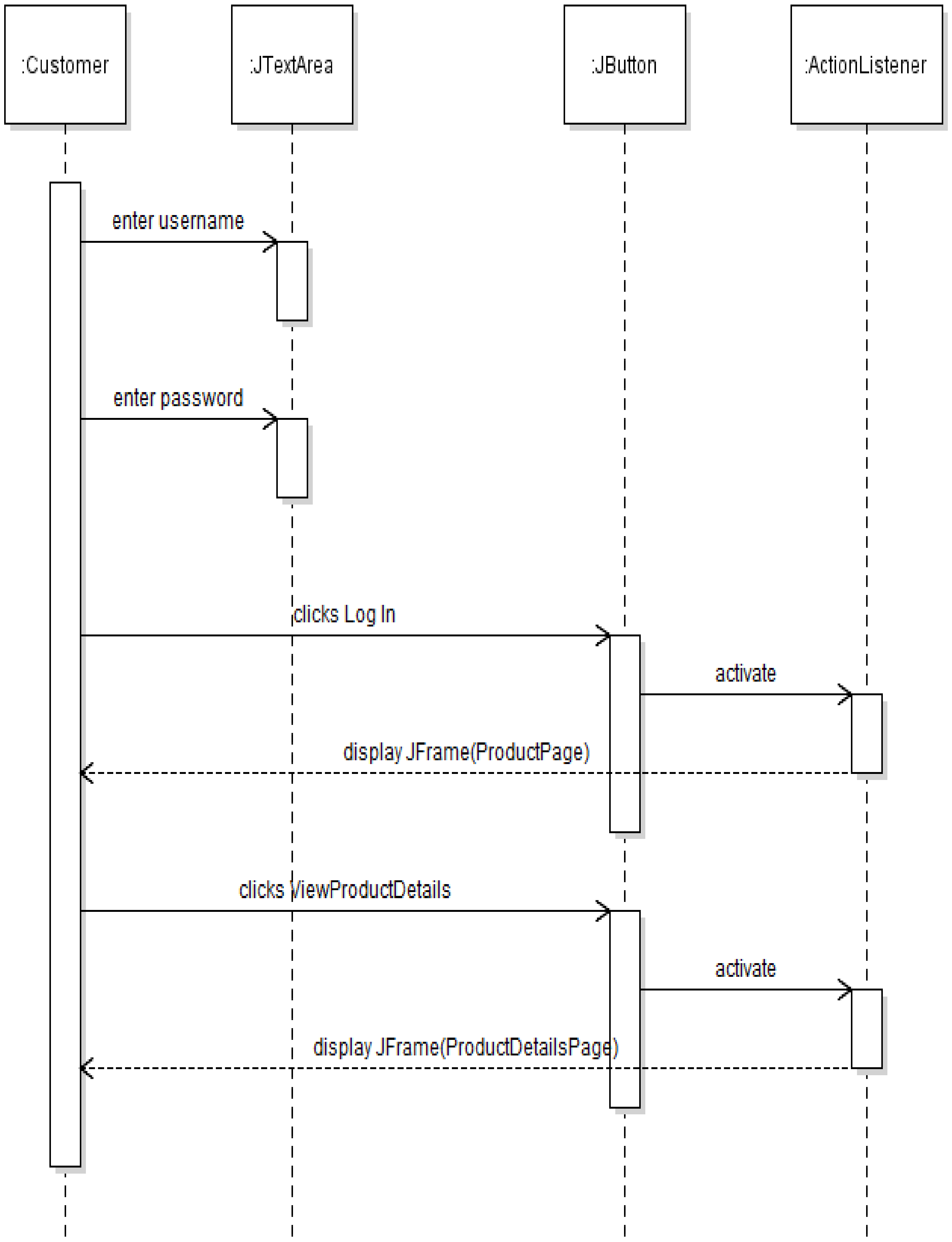
1. Log In



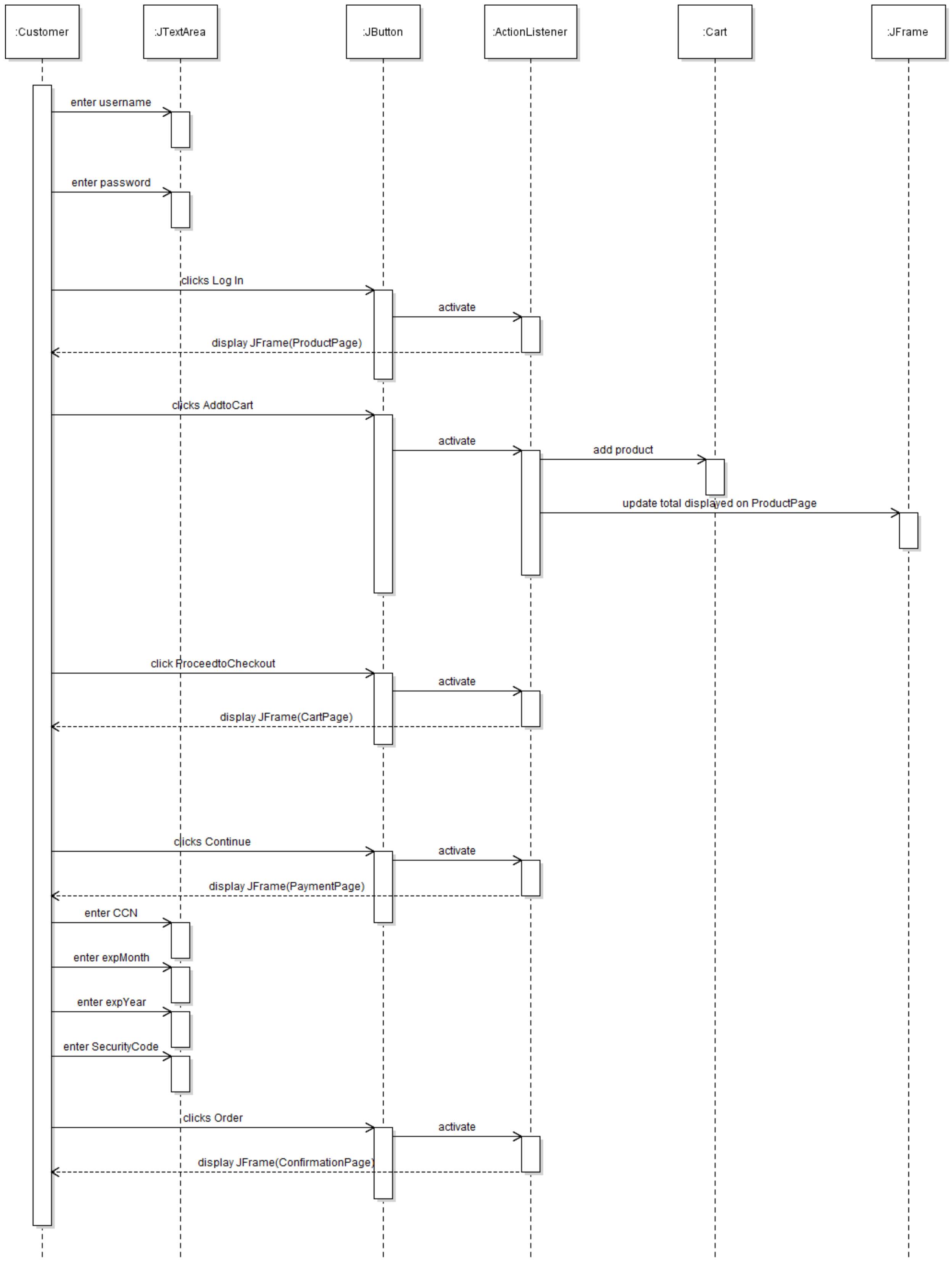
2. Add to Cart


:Customer
:JTextArea
:JButton
:ActionListener
:Cart
:JFrame
enter username
enter password
clicks Log In
activate
display JFrame(ProductPage)
clicks AddtoCart
activate
add product
update total displayed on ProductPage

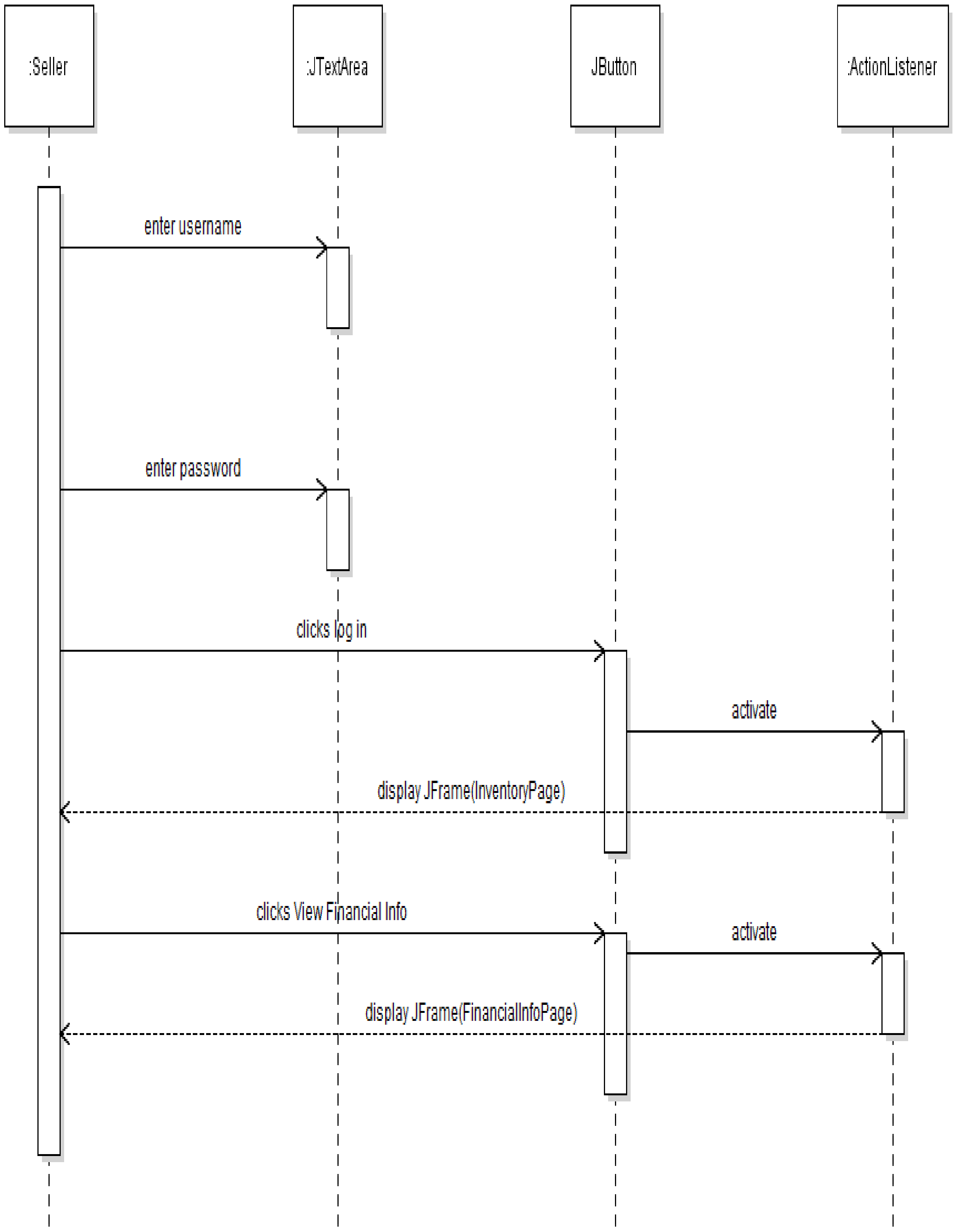
3. Review Product Details



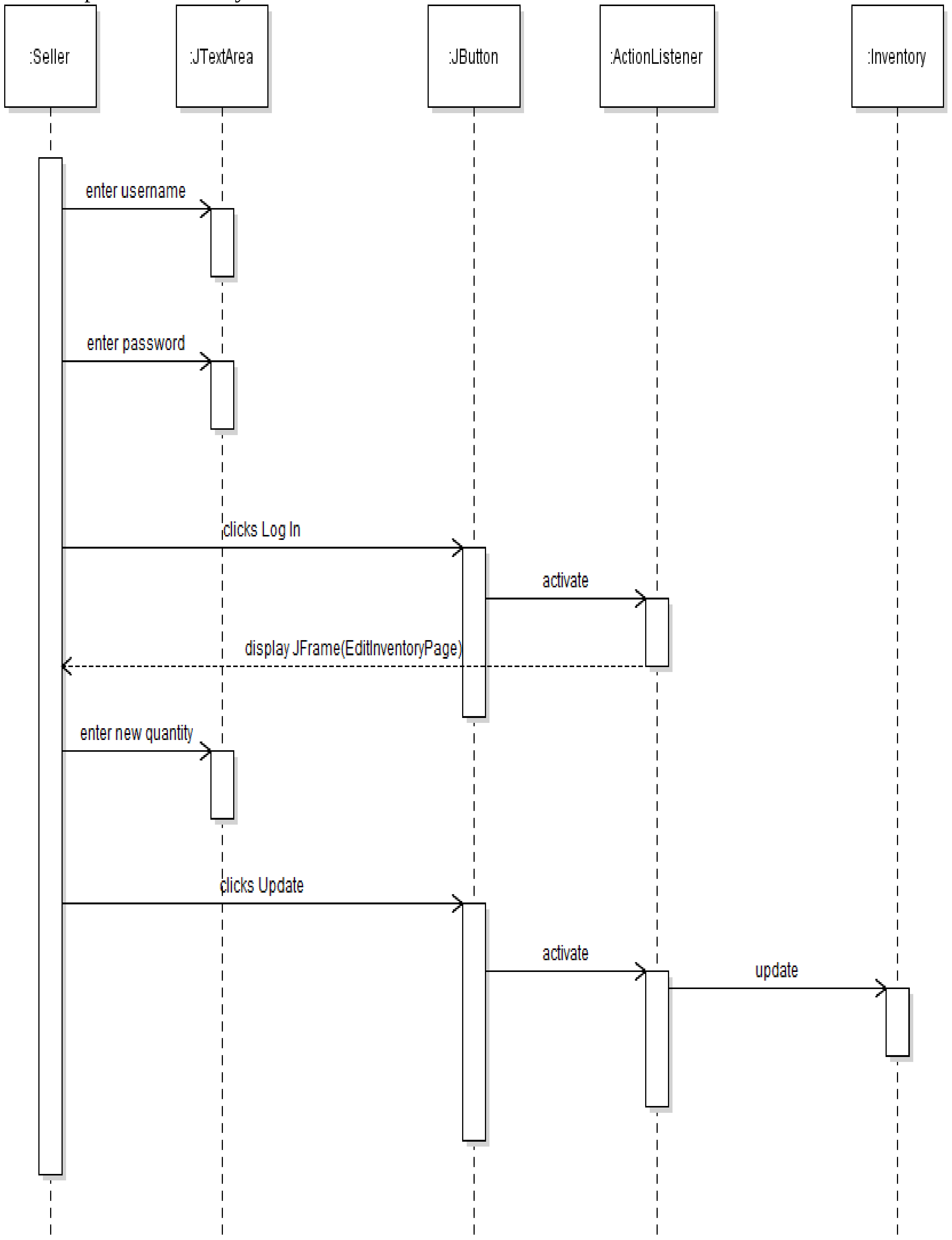
4. Successful Checkout



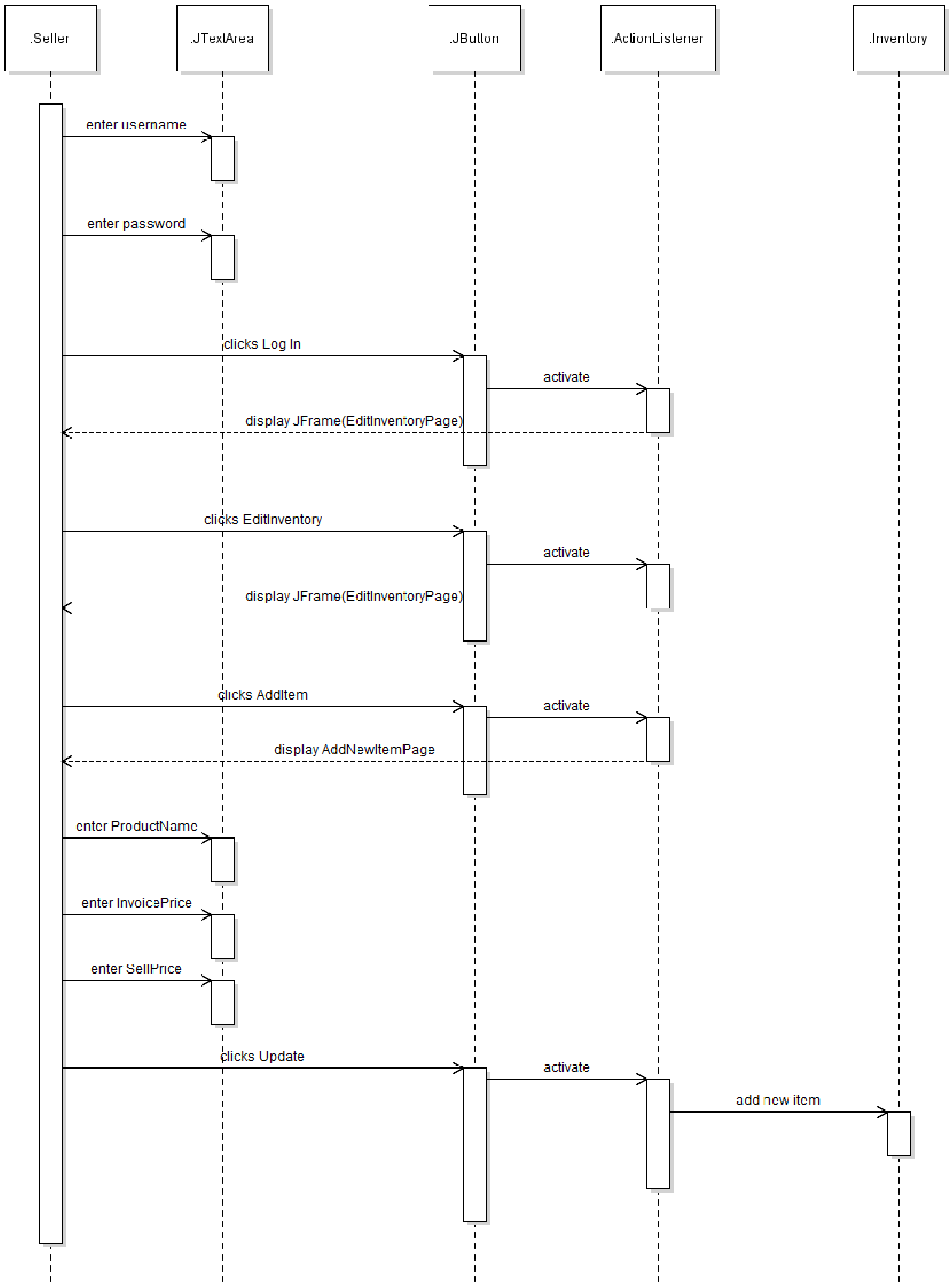
5. Seller Reviews Financial Info



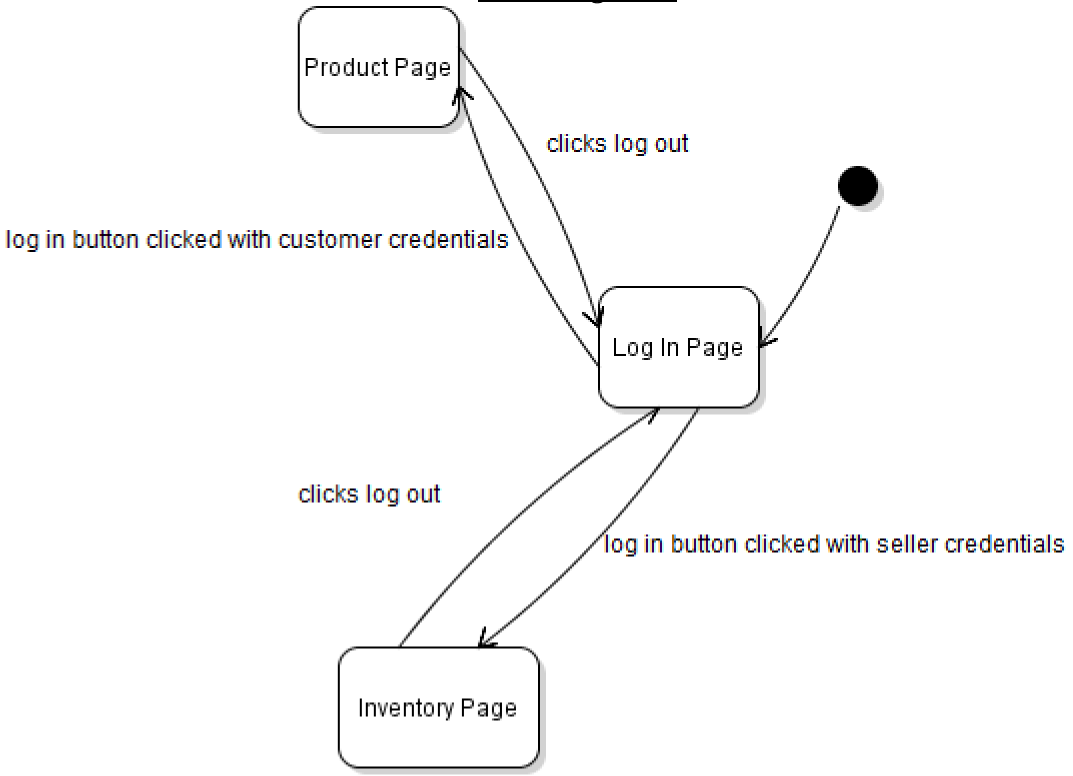
6. Seller Updates Inventory

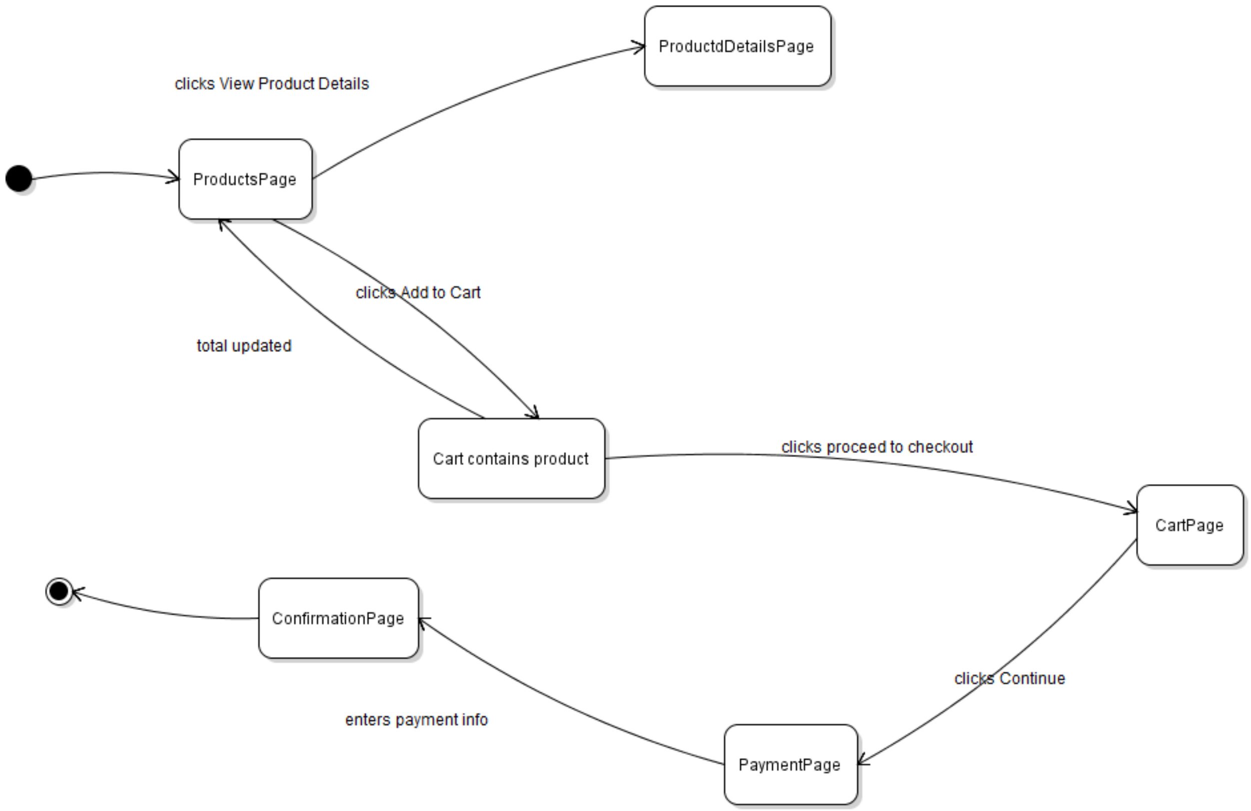


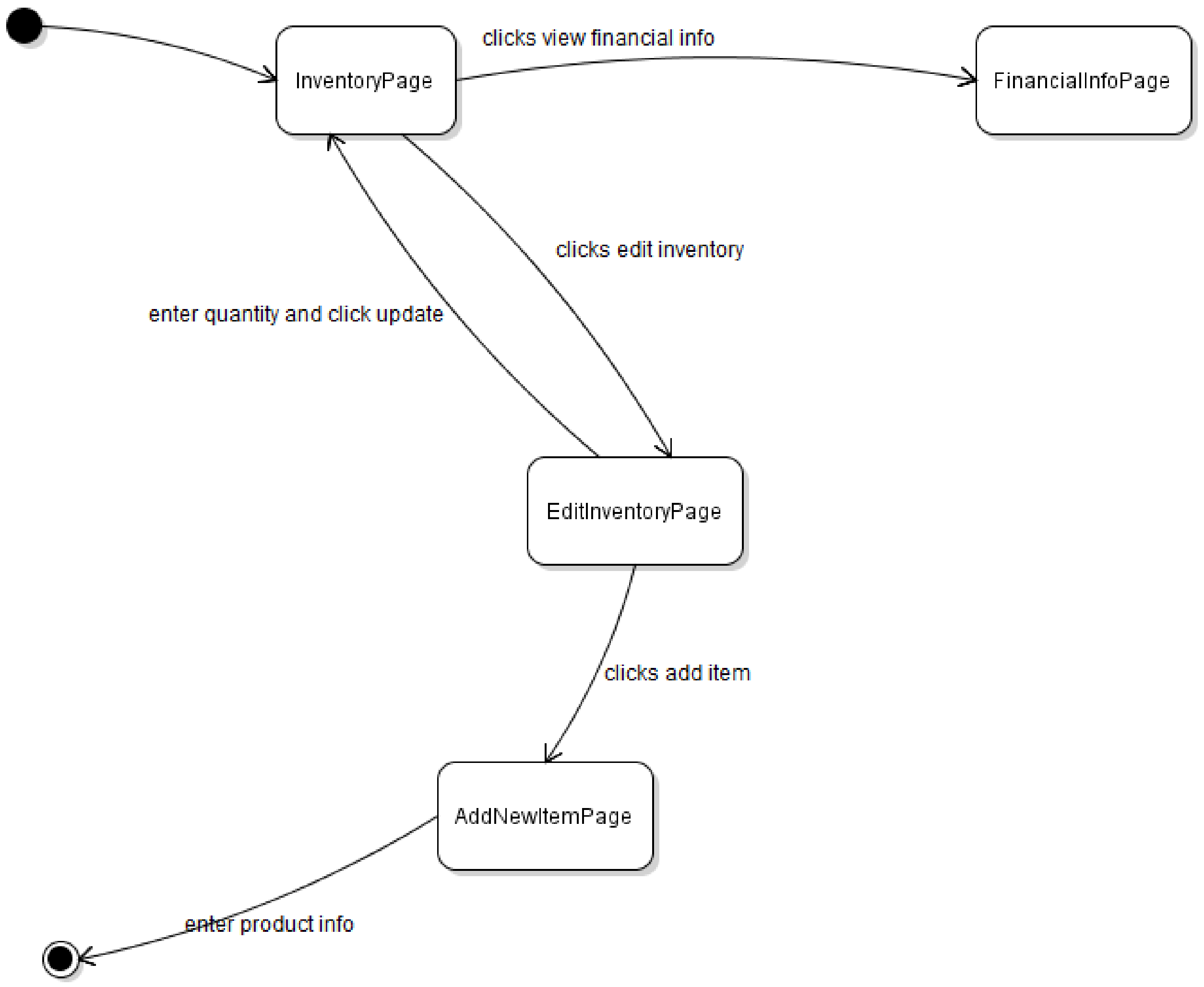
7. Seller Adds Item



State diagrams:


Product Page
clicks log out
log in button clicked with customer credentials
Log In Page
clicks log out
log in button clicked with seller credentials
Inventory Page




InventoryPage
clicks view financial info
FinancialInfoPage
clicks edit inventory
enter quantity and click update
EditInventoryPage
clicks add item
AddNewItemPage
enter product info