

Julian-Justin Djoum

Ross Parsons

Matthew Taylor

Dr. Ionut Cardei

COP4331

30 October 2020

Functional Specification

The Shopping Cart application will allow customers to buy products being sold by sellers in the marketplace. The application will be used by both customers and sellers. Customers will be able to select products to view and purchase, while the sellers will be able to provide the products to sell to the customers. The application client will be built using the Java Swing library as a desktop application, and will interact with Java web services to communicate data to and from the application.

Once the application has opened, a login window (JFrame) is displayed to the current user. The current user may be either a customer or a seller, with each having separate functionality and features used in the application.

In the case of a customer logging in, a window is displayed that shows all products currently listed for sale. The user can select each of the products individually to get more detailed information about the product including a description, availability, and pricing displayed in a separate window. From this separate window, the customer can add the product to the cart or exit out of the window to return to viewing all products. When the customer is ready to checkout, the customer can access the cart in a separate window which displays all of the selected products. Here the customer can remove specific products or quantities of specific products from the cart if needed and then proceed to checkout. The checkout window opens

and the customer is prompted to enter payment information for the order. Order pickup or delivery is not covered within the application. Once checked out, the customer is shown a receipt of the order in a separate window, displaying products ordered, their individual prices, and the total price of the order. The cart is subsequently emptied of all products and the customer is returned to the products window where shopping can be continued, or the customer can exit the application. The products purchased by the customer are each updated in quantity on the products page available to reflect the purchase.

In the case of a seller logging in, a window is displayed that shows all of the products currently listed for sale by the seller. The seller can individually select each product to view more detailed information in a new window, including product ID, type, name, description, quantity available, invoice price, and sell price. The seller can update any of this information from the window to reflect any changes if necessary, such as quantity available or completely canceling the listing, and close out the window to move back to the inventory window. The seller can also select a create listing option from this window, where a new product can be listed for sale. When this option is selected, a new window is opened that collects the product ID, type, name, description, quantity available, invoice price, and sell price. Once the seller has entered in all relevant information, the seller can save this new listing and the inventory is updated to reflect it back on the main listings window. Also from the main window, the seller can access a financial window where various financial statistics are displayed to the seller. This information includes costs, revenues, and profits associated with the products being sold by the seller. This information is displayed in monthly statements, where the seller can access financial information on a monthly basis. The financial window will also show a summary of the last three months of sales, with the differences in costs, revenues, and profits between months being displayed to the seller.

Use Cases

Use Case #1: **User (Customer) Logs In**

1. Application is launched.
2. System displays login window which is a JFrame that contains a JLabel to signify that this is the login window, two text fields (username and password), and a JButton ('Submit')
3. Customer enters username in the first text field.
4. Customer enters password in the second text field.
5. Customer presses the Submit button.
6. Systems displays the shopping catalogue (a JFrame).

Variation #1: **User (Seller) Logs In**

1. This scenario starts in UC1, Step 5 when the user presses the Submit button.
2. System displays the current state of the inventory in a JFrame with a JLabel (showing that user is logged in as a seller) with options to edit listings

Variation #2: **Login Fails**

1. This scenario starts in UC1, Step 5 when the customer presses the Submit button.
2. The system displays a JFrame indicating that the login failed. The JFrame has a JLabel (showing "Login Failed") and a JButton (labeled "OK").
3. Customer presses the OK button and JFrame closes.
4. Customer attempts to login again on the login JFrame

Use Case #2: **Customer Adds Products to Shopping Cart**

1. Customer logs in successfully.
2. A JFrame of the shopping catalogue is displayed by the system. The JFrame contains cards where each card is a product. Each card will have a JButton (labeled "Add to Cart") and another JButton (labeled "Description").
3. Product is added to the shopping cart by clicking the JButton "Add to Cart".

Variation #1: **Product Not Available**

1. This scenario starts in UC2, Step 2.
2. The product that the customer would like to add to the shopping cart is unavailable. A JLabel (labeled "Unavailable" will be in place of the "Add to Cart" JButton).
3. Since there is no JButton to add the product to cart, the customer will not be able to take any action.

Variation #2: **Customer Adds Products to Shopping Cart from Pop-Up Window**

1. This scenario starts in UC2, Step 2 when the catalogue JFrame is displayed by the system.
2. Customer presses the JButton (labeled "Description").
3. Pop-up window displays product information in the form of a JFrame. The JFrame containing the product description has a JButton (labeled "Add to Cart")
4. Product is added to the shopping cart by clicking the JButton "Add to Cart."

Variation #3: **Product Unavailable from Product Description**

1. This scenario starts in UC2, Step 2, when the catalogue JFrame is displayed by the system.
2. Customer presses the JButton (labeled "Description").
3. Pop-up window displays product information in the form of a JFrame. The JFrame containing the product description has a JLabel (labeled "Unavailable")
4. Since there is no JButton to add the product to cart, the customer will not be able to take any action.

Use Case #3: **Customer Reviews Product Details**

1. This scenario starts in UC1 Step 6, after the customer has successfully logged in and a JFrame of the shopping catalogue is displayed by the system.
2. User clicks on the JButton labeled "Description".
3. A JFrame showing product details is displayed by the system.
4. The JFrame of the product details shows various fields such as price, a picture of the product, a text field with a written description of the product, etc.

Use Case #4: **Customer Checks Out**

1. Customer is presented with a JButton to "Check Out" once they add at least 1 product to their cart, which is a JList
2. Customer enters payment information in required fields.
3. Customer is presented with a receipt showing details of the transaction

Variation #1: **Customer Continues Shopping**

1. This scenario starts in UC4 Step 1, the customer is shown the JButton to "Continue Shopping" and check out later, once they add a product to their cart, saving what is currently in their cart

Variation #2: **Error in Payment**

1. This scenario starts in UC4 Step 2, the customer is presented with a JLabel message containing "Error in making payment."
2. Customer reattempts payment or exits checkout to catalogue JFrame

Use Case #5: **Seller Reviews Inventory**

1. This scenario starts in UC1, Variation1, step 2 where the System displays the current state of the inventory in a JFrame with a JLabel (showing that user is logged in as a seller) with options to update inventory (add products, see available quantity, specifying product name, see the invoice price, update available quantity, etc.). Each of these options will be labeled JButtons.
2. The seller presses the JButton labeled "Review Inventory."
3. The system displays the current inventory of each listing in a new JFrame.

Variation #1: **Seller Updates Inventory**

1. This scenario starts in UC5, Step 3 where the seller is presented with the current inventory of the selected product.
2. The seller presses the JButton labeled "Update Inventory".
3. A new JFrame is displayed with an option to update the quantity of the product to desired quantity.

Use Case #6: **Seller Adds New Product**

1. This scenario starts in UC5, Step 1 where the seller is presented with a group of JButtons highlighting the options the seller can take.
2. The seller selects the 'Add New Listing' option
3. The system prompts the seller for new listing information
4. The seller adds the new product information to the listing
5. The system prompts: "Confirm the displayed inventory update."
6. The system updates the inventory to display the new product.

Use Case #7: **Seller Reads About Revenue, Sales, & Profit**

1. The seller is presented the option to view business details when logged into the system by pressing a JButton called "Business Details"
2. The system then shows the seller their total revenue, number of sales, and total profit in a JLabel

Variation #1: **Seller Has No Revenue, Sales, or Profit**

1. This scenario starts in UC7, Step 2 where the system displays financial data
2. The system displays "No financial data available."

Glossary

- Transaction
 - A customer transfers the cost of the items they want to purchase into the system. The seller receives the money, gives the customer their items, and shows the customer a receipt for their purchase.
- Product
 - An item with a price that the seller has listed for sale.
- Seller
 - A user of the application with products available for sale in the system to buyers. Has special privileges to view business details which include revenue, sales, and profit.
- Customer
 - A user of the application with purchasing privileges that can purchase items listed for sale by sellers.
- JFrame
 - Application window where data throughout the application will be displayed.
- JButton
 - A button on the screen that is displayed within the JFrame. Buttons will cause data to be displayed or changed within the application.
- JLabel
 - A label which can display various types of data to the user. It is used to display data within a window or can be used as an alert.
- Field
 - A text box within the frame that collects data from a customer or seller.
- List
 - A list of data within the frame.

Classes:

Customer

- CRC Card

- Browses all products
- Views individual product details
- Adds products to cart
- Checks out with cart
- Enters payment
 - **Relations:**
 - **Order**
 - **Cart**
- UML Diagrams
 - Attributes
 - id
 - Name
 - Cart

Seller

- Adds products for sale
- Updates product inventory
- Reviews product inventory
- View SellerReport
 - **Relations:**
 - **Product**
 - **SellerReport**
 - UML Diagrams
 - Attributes
 - id
 - name
 - cart

Order

- Manages the current amount of items and current price total of a customer's cart
- Displays of the amount of items a customer buys
- Displays the total cost of a customer's order
- Calculates total order price
- Allows a customer to check out
- Prints receipt to customers
 - **Relations:**
 - **Customer**
- UML Diagrams
 - Attributes
 - id
 - products
 - timestamp

Product

- Displays individual product details (name, price, units)
- Displays individual product reviews
- Manages current product units

- Allows Seller to edit individual product details (name, price, units)
 - **Relations:**
 - **Seller**
- UML Diagrams
 - Attributes
 - id
 - seller_id
 - Name
 - description
 - Quantity
 - price

SellerReport

- Displays the seller's revenue, sales, and profit if (sales \geq 1)
- Displays that the seller has no revenue, sales, and profit (sales $<$ 1)
- Displays the seller's current inventory
 - **Relations:**
 - **Seller**
 - **Product**
- UML Diagrams
 - Attributes
 - seller_id
 - products
 - Timestamp

Cart

- Keeps a customer's chosen items in store
- Updates inventory for bought products
 - Relations:
 - Customer
 - Product
- UML Diagrams
 - Attributes
 - Customer_id
 - productsList
 - total
 - Timestamp

CRC Cards

Customer
<ul style="list-style-type: none">• Browses all products• Views individual product details• Adds products to cart• Checks out with cart• Enters payment
Relations: <ul style="list-style-type: none">• Order• Cart

Seller
<ul style="list-style-type: none">• Adds products for sale• Updates product inventory• Reviews product inventory• Access to view SellerReport
Relations: <ul style="list-style-type: none">• Product• SellerReport

Order
<ul style="list-style-type: none">• Manages the current amount of items and current price total of a customer's cart• Displays of the amount of items a customer buys• Displays the total cost of a customer's order• Calculates total order price• Allows a customer to check out• Prints receipt to customers
Relations: <ul style="list-style-type: none">• Customer

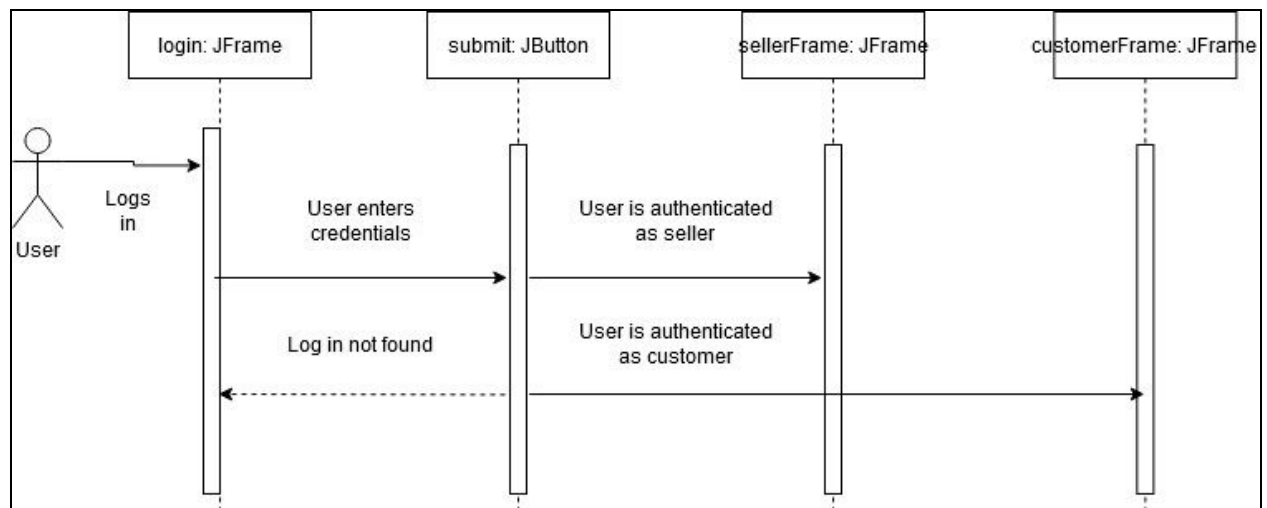
Product
<ul style="list-style-type: none">• Displays individual product details (name, price, units)• Displays individual product reviews• Manages current product units• Allows Seller to edit individual product details (name, price, units)
Relations: <ul style="list-style-type: none">• Seller

Cart
<ul style="list-style-type: none">• Keeps a customer's chosen items in store• Updates inventory for bought products
Relations: <ul style="list-style-type: none">• Customer• Product

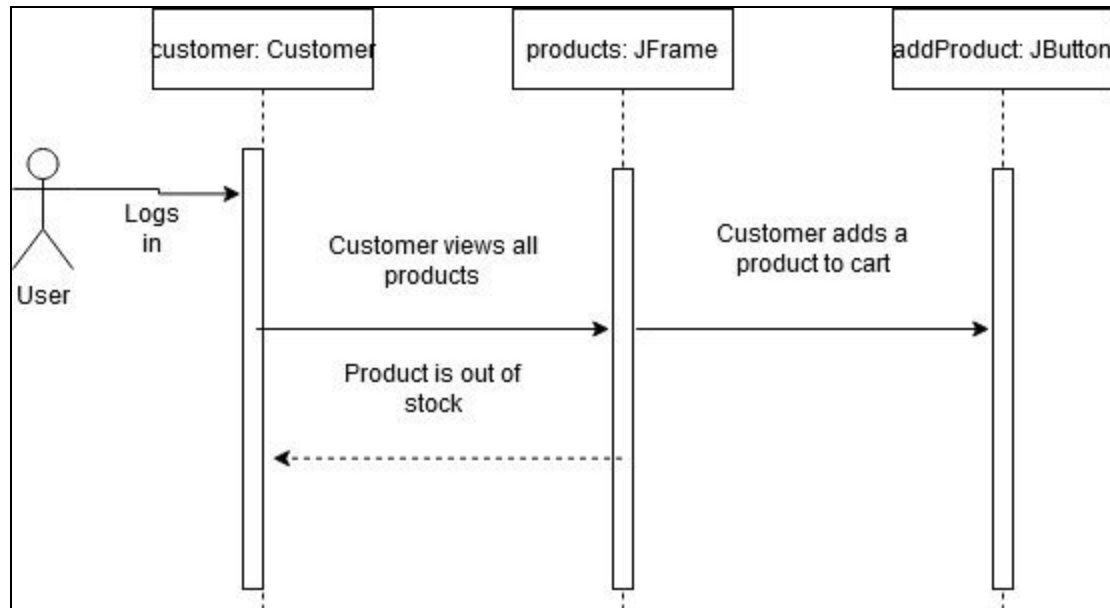
SellerReport
<ul style="list-style-type: none">• Displays the seller's revenue, sales, and profit if (sales \geq 1)• Displays that the seller has no revenue, sales, and profit (sales $<$ 1)• Displays the seller's inventory
Relations: <ul style="list-style-type: none">• Seller• Product

Sequence Diagrams

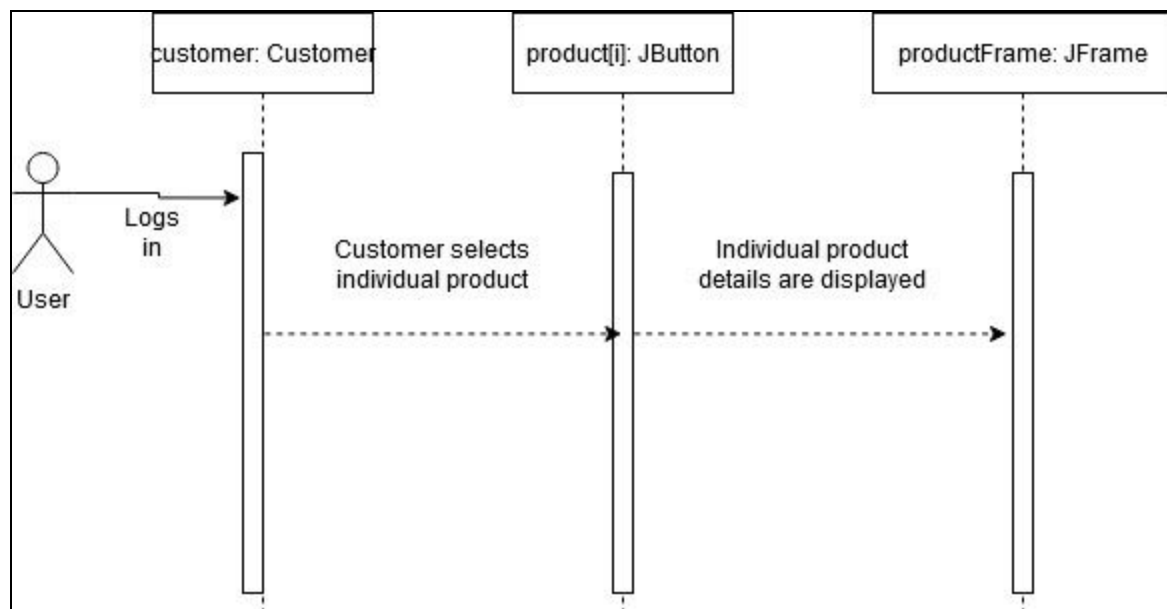
Use Case #1: **User (Customer) Logs In**



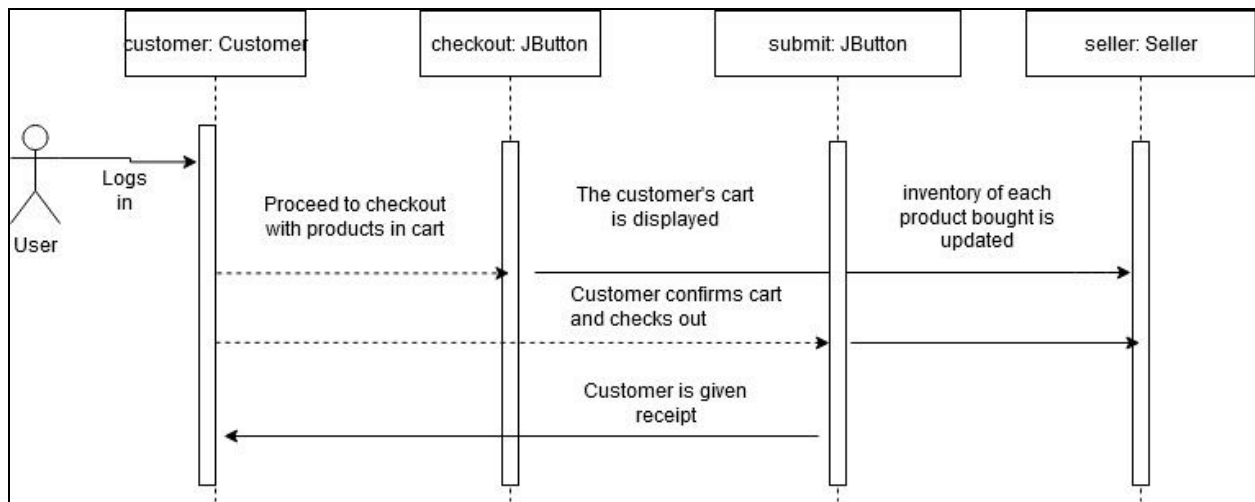
Use Case #2: Customer Adds Products to Shopping Cart



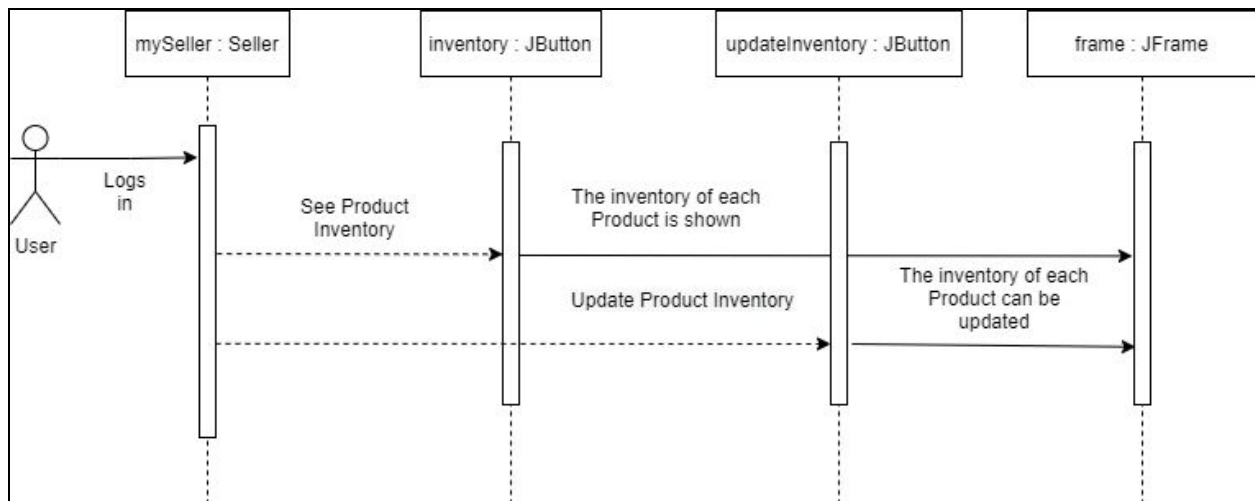
Use Case #3: Customer Reviews Product Details



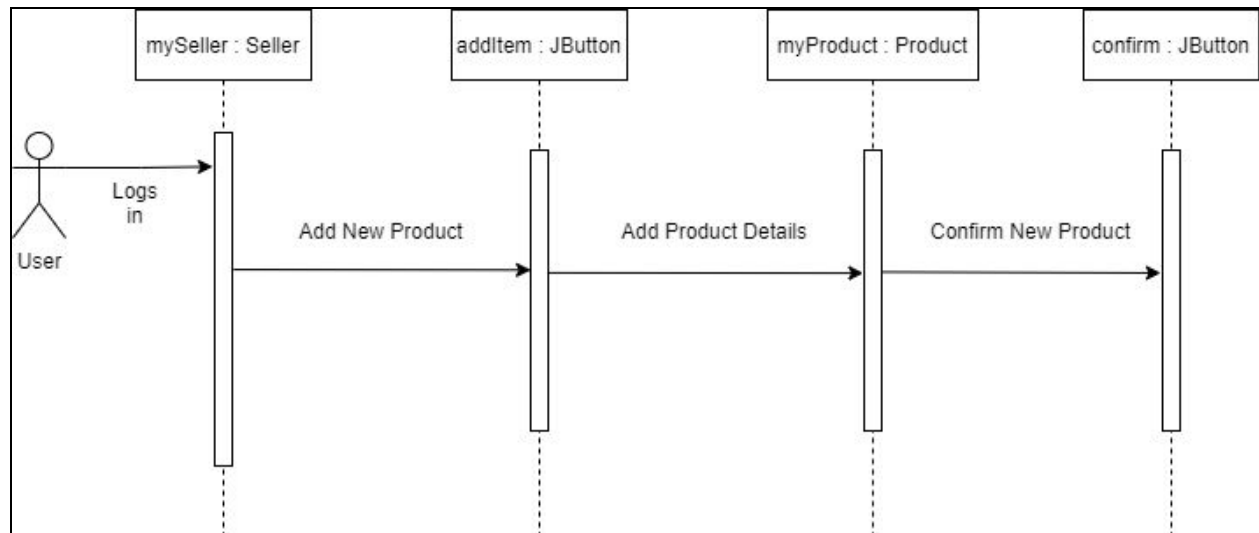
Use Case #4: Customer Checks Out



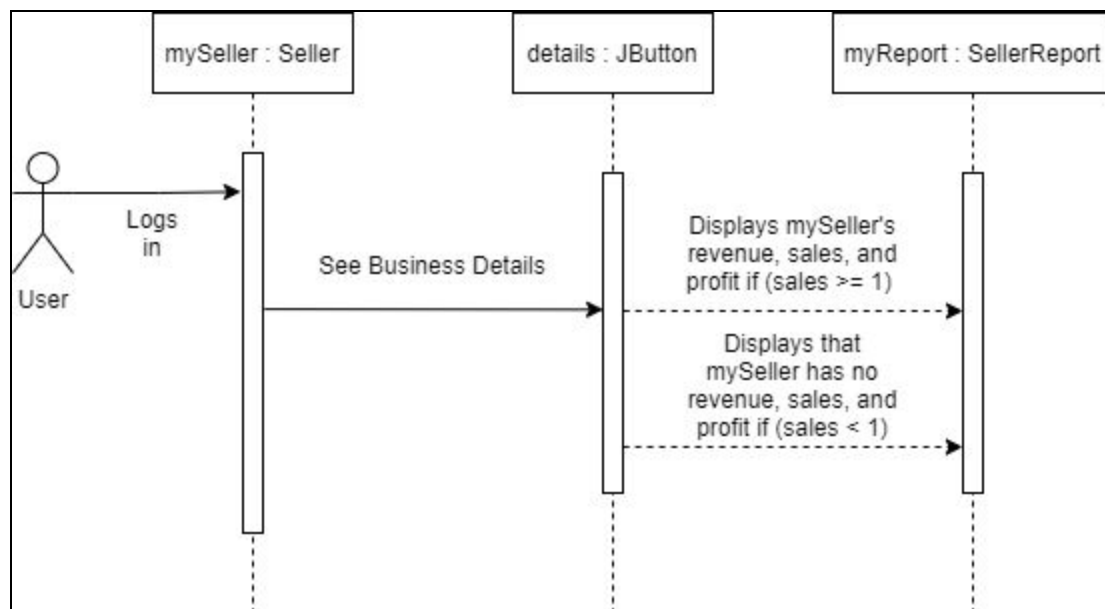
Use Case #5: Seller Reviews Inventory



Use Case #6: Seller Adds New Product



Use Case #7: Seller Reads About Revenue, Sales, & Profit



Class Diagram

