Automated Smart Shopping Cart

Team 35 - André Morais, Miguel Rodrigues, Paul Rodrigues

The idea is to develop the architecture of an automatic checkout system that makes shopping easier. It tracks items as you add them to your cart and updates your total in real-time. You can see what's in your cart and how much you've spent anytime. When you're done, just confirm the payment and leave, no waiting in line.

Structure Diagram

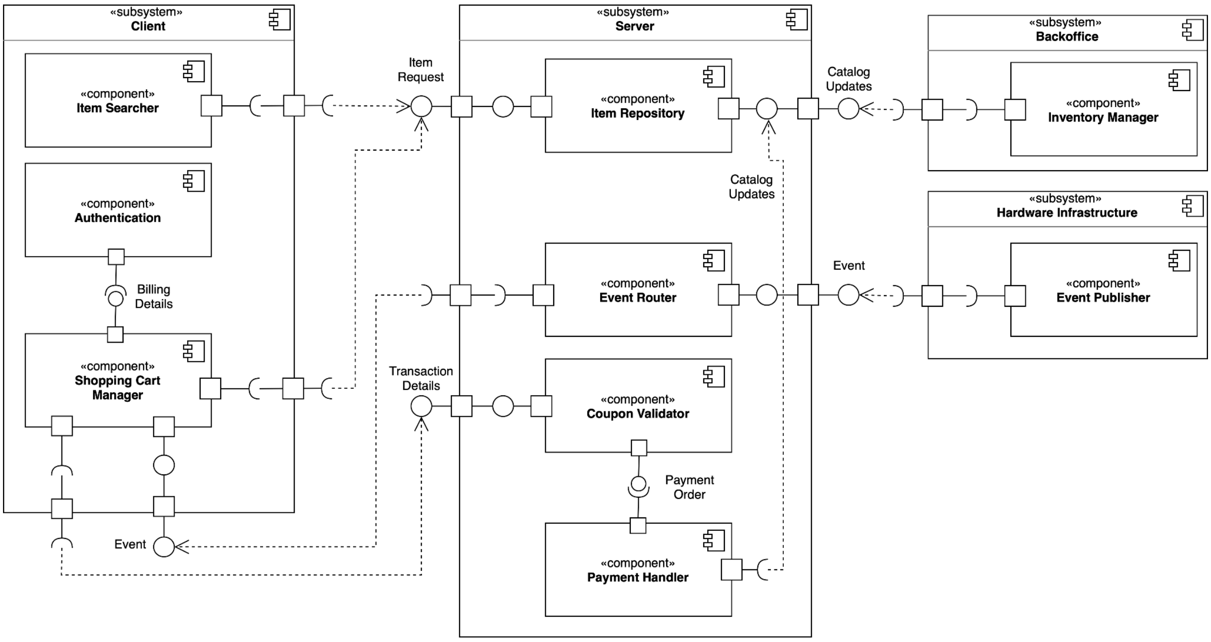

Fig. 1 – Automated Smart Shopping Cart components as a structure diagram.

Client Subsystem:

- Item Searcher: This component allows users to search for items. It sends item requests to the Item Repository.
- Authentication: Manages user authentication and billing details.
- Shopping Cart Manager: Keeps track of items added or removed from the cart, displays the running total, and maintains the list of items in the cart. It also handles transaction details and interacts with the Coupon Validator for discounts.

Server Subsystem:

- Item Repository: Stores information about available items, handles item requests from the client and receives catalog updates from the Inventory Manager and Payment Handler.
- Event Router: Acts as a broker of all the events generated by the system, such as item additions/removals. It routes them to the appropriate components, such as the Shopping Cart Manager.

- Coupon Validator: Validates electronic coupons and applies discounts to transactions before sending payment orders to the Payment Handler.
- Payment Handler: Handles payment processing, supporting various payment methods like credit/debit cards, Venmo, etc. It also communicates with the Item Repository for transaction updates.

Backoffice Subsystem:

- Inventory Manager: Manages store inventory and ensures accurate stock information, as a consequence it is capable to lookup at each item's location.

Hardware Infrastructure Subsystem:

- Event Publisher: Detects events related to cart interactions, such as item additions/removals, and publishes them to the Event Router for processing.

Sequence Diagrams

Adding an item to the cart

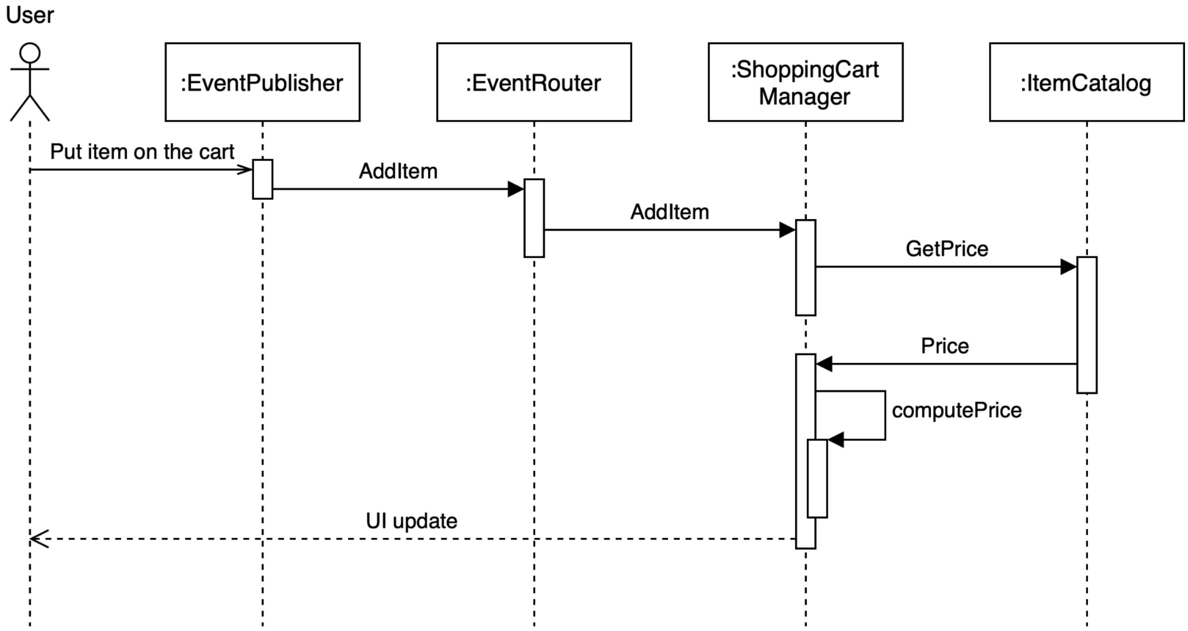

Fig. 2 – Sequence diagram for adding an item to the cart.

1. A user places an item in the cart.
2. The Event Publisher detects the addition of the item to the cart.
3. Event Publisher notifies the Event Router about the addition of events.
4. Event Router routes the event to the Shopping Cart Manager.
5. The Shopping Cart Manager requests the price of the added item from the Item Catalog.
6. Item Catalog provides the price information to the Shopping Cart Manager.
7. ShoppingCartManager updates the UI to reflect the addition of the item and its price.

Leaving the store

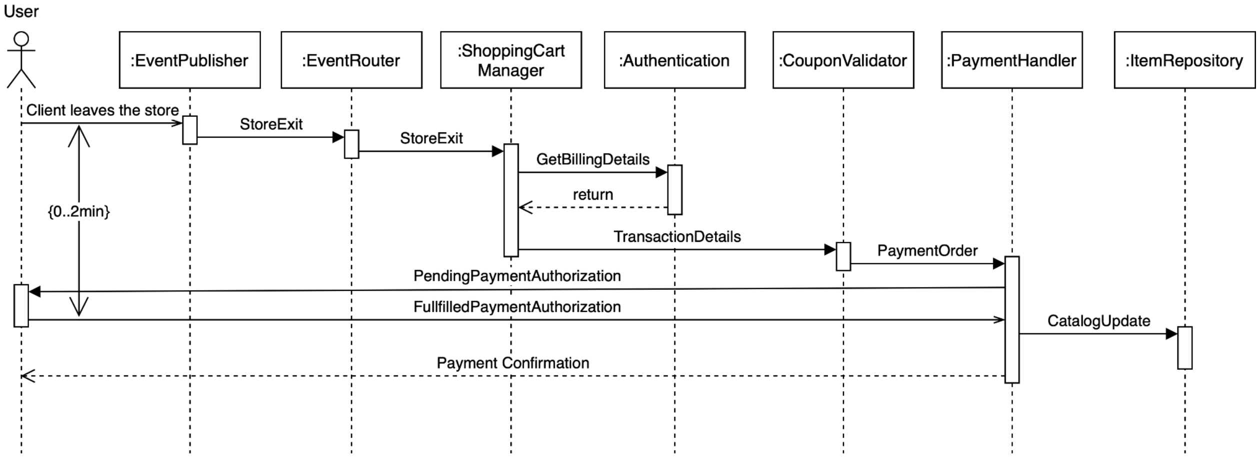

Fig. 3 – Sequence diagram for when a User leaves the store.

1. The user leaves the store with the cart.
2. The Event Publisher detects the store exit event.
3. Event Publisher notifies the Event Router about the store exit.
4. Event Router routes the event to the Shopping Cart Manager.
5. The Shopping Cart Manager requests billing details from the Authentication component.
6. Authentication verifies the user's identity and returns the billing details to the Shopping Cart Manager.
7. The Shopping Cart Manager sends transaction details to the Coupon Validator.
8. Coupon Validator generates a payment order and forwards it to the Payment Handler.
9. The payment Handler processes the payment order and sends the user a pending payment authorization request.
10. The user fulfills the payment authorization within the specified time limit (2 minutes).
11. Upon receiving the fulfilled payment authorization, the Payment Handler confirms the payment and updates the Item Repository with the transaction details.
12. Payment Handler notifies the User of the payment confirmation.

Main use cases

From the assignment description we can tailor a few use cases, such as:

1. Adding items to the cart
2. Removing items from the cart
3. Viewing cart contents and total price
4. Handling payments and membership benefits

Use Case 1: Adding Items to the Cart

Actors:

- User, Cart System

Description:

- Users approach the Automated Smart Shopping Cart system with their carts.
- Users begin selecting items from the shelves and placing them in their carts.
- The cart system detects each item as it is placed in the cart and updates the running total.
- Users continue adding items until they have completed their shopping.

Preconditions:

- The cart system is operational and available for use.
- User has entered the store and obtained a cart.

Postconditions:

- Items are successfully added to the cart.
- The cart system updates the running total to reflect the newly added items.

Use Case 2: Removing Items from the Cart

Actors:

- User, Cart System

Description:

- User approaches the system with his cart containing items.
- User reviews the items in his cart and decides to remove one or more items.
- User takes them out of the cart.
- The cart system detects the removal of each item and updates the running total accordingly.

Preconditions:

- User has already added items to his carts.
- The cart system is operational and available for use.

Postconditions:

- Items are removed from the cart.
- The cart system updates the running total to reflect the removal of the items.

Use Case 3: Viewing Cart Contents and Total Price

Actors:

- User, Cart System

Description:

- User approaches the system with his carts containing items.
- User interacts with the system to view the contents of his carts.
- The cart system displays a list of items currently in the cart, along with their quantities and prices.
- User reviews the list of items to ensure everything is correct.
- User checks the running total displayed by the cart system to see the total price of the items in his cart.

Preconditions:

- User has already added items to his cart.

- The cart system is operational and available for use.

Postconditions:

- User successfully views the contents of his cart and the total price of the items.
- The cart system provides accurate information about the items in the cart and the total price.

Use Case 4: Handling Payments and Membership Benefits

Actors:

- User, Cart System, Payment Handler, Membership Database

Description:

- The user approaches the system with his cart containing items.
- The user reviews the items in his cart and confirms their intention to proceed to checkout.
- The cart system prompts the user to initiate the payment process.
- The user selects his preferred payment method, if not selected already in the application setup.
- The cart system securely transmits the payment information to the Payment Handler for verification.
- The Payment Handler validates the payment information and processes the payment transaction.
- If the user has associated membership benefits, the cart system checks the Coupon Validator to apply any available discounts or offers.
- The system updates the total price to reflect any applied membership benefits.
- The user receives confirmation of the completed payment transaction.

Preconditions:

- The user has already added items to their cart.
- The user has verified the contents of their cart and the total price.
- The cart system is operational and available for use.
- The user has valid payment information and may have associated membership benefits.

Postconditions:

- Payment for the items in the cart is successfully processed.
- Any applicable membership benefits are applied to the transaction.
- The user receives confirmation of the completed payment transaction in the terminal and application.

Quality Attributes

Quality Attribute	What does it mean?	Acceptance level?	How important?
Usability	Intuitive, user-friendly, accessible	Users should be able to navigate the system intuitively and perform tasks without confusion.	A usable interface enhances user satisfaction and reduces the likelihood of errors, leading to increased customer loyalty.
Reliability	Consistency, Fault Tolerance	The system should operate without failures or unexpected downtime, providing a dependable shopping experience.	Reliability is crucial for building trust with users and ensuring smooth operations, minimizing disruptions that could lead to lost sales or negative customer experiences.
Security	Protection, Encryption	Data should be encrypted and access restricted to authorized users only to prevent data breaches and fraud.	Security breaches can lead to financial loss, reputational damage, and legal liabilities. Ensuring robust security measures is essential for protecting users and the business.
Scalability	Flexibility, Adaptability	The system should be able to scale up to accommodate growing user demands and transaction volumes.	While scalability is crucial for accommodating growth, immediate scalability may not be as critical as other attributes such as reliability and security. However, it becomes increasingly important as the user base expands over time.

Quality Attribute	What does it mean?	Acceptance level?	How important?
Performance	Responsiveness, Efficiency	The system should respond promptly to user actions and process transactions quickly.	Performance directly impacts user satisfaction and perception of the system's reliability. Slow response times or delays can lead to abandoned transactions, negatively affecting the user experience.
Maintainability	Modular, Upgradable	The system should have modular components and well-documented code to facilitate maintenance and updates.	Maintainability influences the system's long-term viability and the ability to adapt to changing business requirements.
Interoperability	Compatible, Integration	The system should support standard protocols and APIs to enable seamless integration with third-party services and future expansion.	Interoperability enables the system to leverage external resources and functionalities, enhancing its capabilities and providing additional value to users.