

Faculdade de Engenharia da Universidade do Porto



Automated Smart Shopping Cart

Software Systems Architecture

Team T24:

Henrique Oliveira Silva up202007242@up.pt

João Pedro Matos Araújo up202007855@up.pt

Rui Filipe Cunha Pires up202008252@up.pt

Pedro Miguel Nunes up202004714@up.pt

Diogo Miguel Ferreira Costa up202007770@up.pt

1. Architecture Overview	2
2. Logical View: Components and Connectors Diagram	2
3. Process View: Sequence Diagrams	3
a) Adding an item to the cart	4
b) Walking out of the Store:	5
4. Use-cases	5
Use-Cases List	5
Use-cases View	6
Use Case 1: User adds items to the cart	6
Use Case 2: User completes the checkout process	7
5. Additional Considerations	8
Quality Attributes	8
Architectural Patterns	9
Architectural Decisions	9

1. Architecture Overview

The Automated Smart Shopping Cart system integrates both hardware and software components to enhance the shopping experience by simplifying the process. It consists of several key components working together to detect items in the cart, display a running total, and facilitate checkout.

2. Logical View: Components and Connectors Diagram

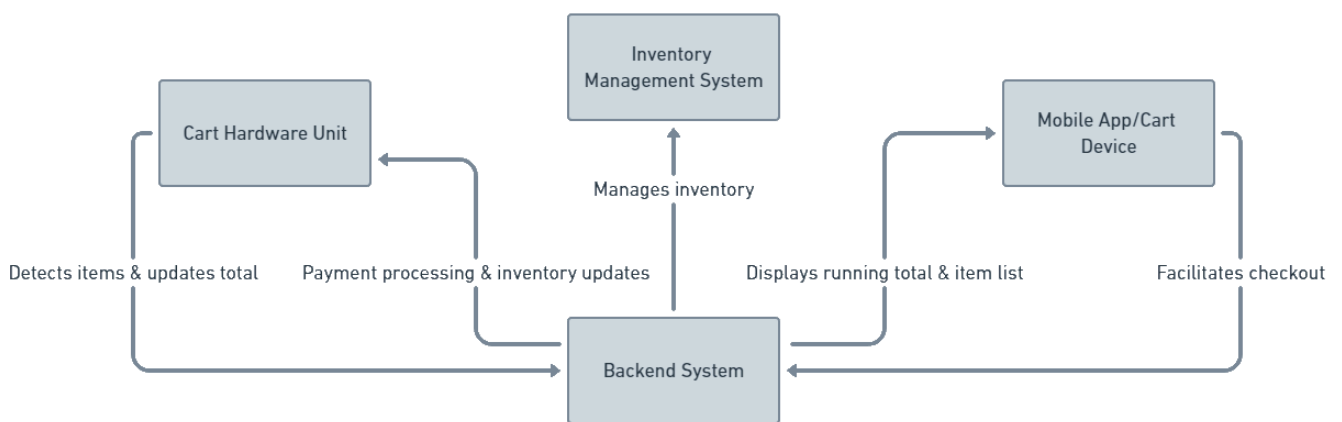


Fig. 1 - Logical View

The system will consist of the following main components:

- **Cart Hardware Unit:** Detects when items are added or removed from the cart and displays a running total.
- **Mobile App or Cart Device:** Provides a user interface for displaying the running total, list of items in the cart, and facilitating checkout.
- **Backend System:** Manages user authentication, item detection, pricing, discounts, and payment processing.
- **Inventory Management System:** Manages store inventory and item lookup functionality.

These components will interact through well-defined APIs (Application Programming Interfaces).

- The Mobile App/Cart Device interacts with the Backend System to display the running total, list of items in the cart, and facilitate checkout. It sends requests for updating the display and processing payments.

- The Backend System interacts with both the Mobile App/Cart Device and the Cart Hardware Unit. It manages user authentication, item detection, pricing, discounts, payment processing, and inventory updates.
- The Cart Hardware Unit detects items in the cart, displays the running total, and prompts for payment if the user is not using a mobile app. It interacts with the Backend System for payment processing and inventory updates.

3. Process View: Sequence Diagrams

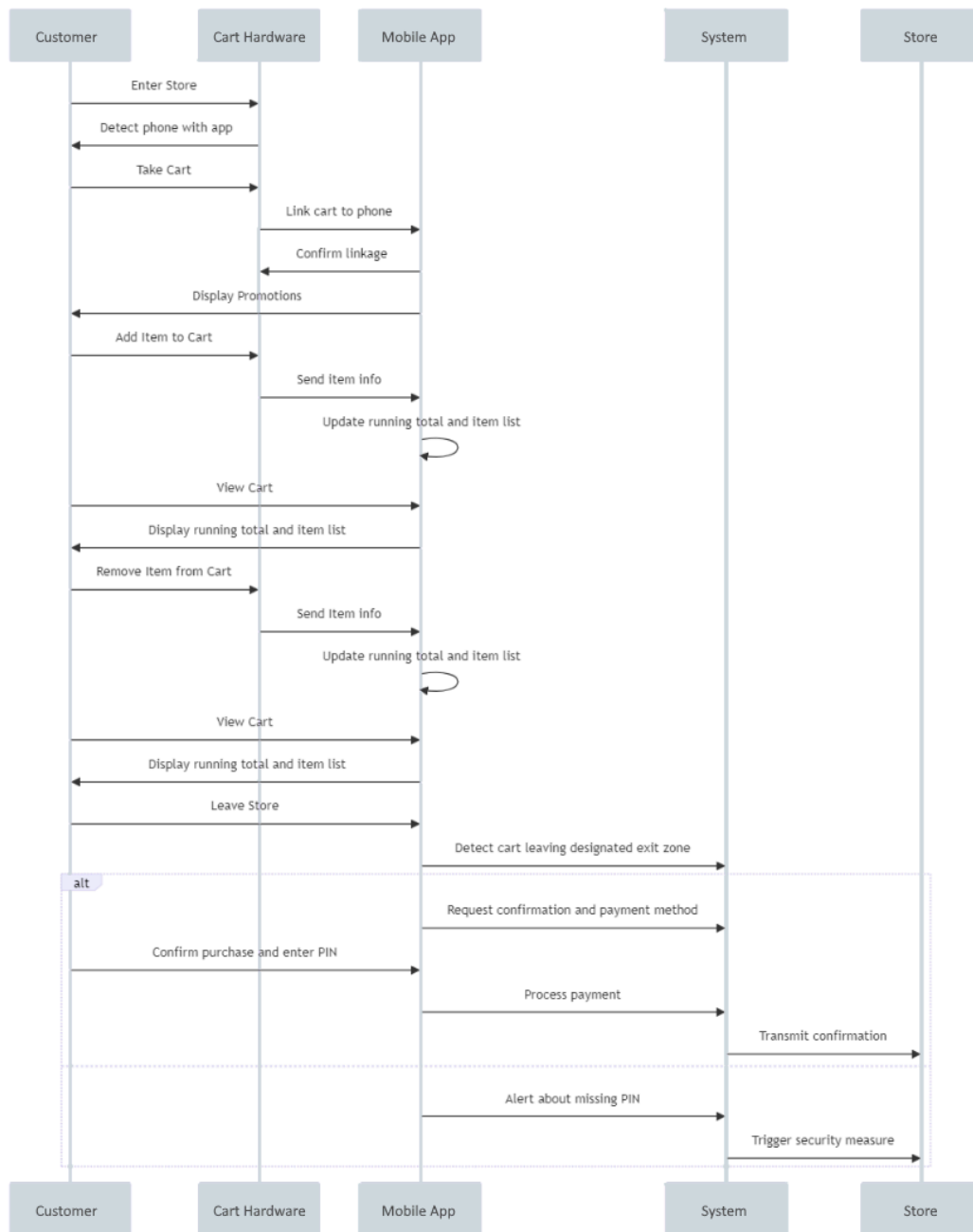


Fig. 2 - Sequence Diagram

Events overview:

a) Adding an item to the cart

1. Store Entry:

- When the customer enters the store, sensors in the cart hardware detect the presence of the customer's phone with the app. This allows for seamless integration of the customer's shopping experience with the mobile app.

2. Cart-Phone Association:

- As the customer takes the cart, the system automatically establishes a connection between the cart and the customer's phone. This association enables personalized features and interactions throughout the shopping journey.

3. Display Promotions:

- The system presents tailored promotions or exclusive offers to customers via the mobile app as they shop. These could include discounts, buy-one-get-one-free deals, coupons, or loyalty rewards. By showcasing enticing promotions, the system encourages customers to explore more products and potentially increase their spending.

4. Item Addition to Cart:

When the customer adds an item to the cart, the cart hardware sends detailed item information, such as the item ID, name, and price, to the mobile app. This ensures accurate tracking and updating of the customer's shopping list.

5. Total and Item List Update:

- In response to item additions or removals, the mobile app promptly updates the running total and the item list in real-time. This provides customers with an accurate overview of their current shopping cart contents and total expenditure.

6. Cart Viewing:

- Customers have the convenience of checking their shopping cart directly on the mobile app. They can review the item list,

quantities, and prices to ensure their shopping needs are met before proceeding to checkout.

7. Item Removal from Cart:

- If a customer decides to remove an item from the cart, they can easily do so through the mobile app interface. This prompts the app to update the total and the item list accordingly, reflecting the changes made by the customer.

b) Walking out of the Store:

1. Store Exit:

- When the customer is ready to leave, the system detects the cart leaving the designated exit zone, signaling the end of the shopping trip.

2. Purchase Confirmation and Payment:

- The mobile app requests purchase confirmation and prompts the customer to select a payment method. The customer confirms the purchase and enters a PIN for authentication.
- If PIN is not entered:
- The mobile app alerts the customer about the missing PIN.
- The system triggers a security measure, possibly sounding an alert or notifying store staff.

3. Payment Processing:

- Upon confirmation, the system securely processes the payment using the selected method, ensuring the transaction's authenticity. Once processed, the confirmation is transmitted to the store's backend system, finalizing the transaction.

These events represent the primary interactions between the customer, the cart, the app, and the system during the shopping process.

4. Use-cases

Use-Cases List

1. User adds items to the cart
2. User completes the checkout process

3. User removes items from the cart
4. User views contents of cart
5. User applies electronic coupons to purchases
6. User views aisle location of items in the store

Use-cases View

Use Case 1: User adds items to the cart

Actors: User

Preconditions: User is logged in or has a cart linked to their device.

Scenario:

1. User adds an item to the cart.
2. Cart hardware detects the item and identifies it.
3. Cart hardware sends item information to the app/display unit.
4. App/display unit receives information and updates

Postconditions: The item is successfully added to the cart, and the running total is updated.

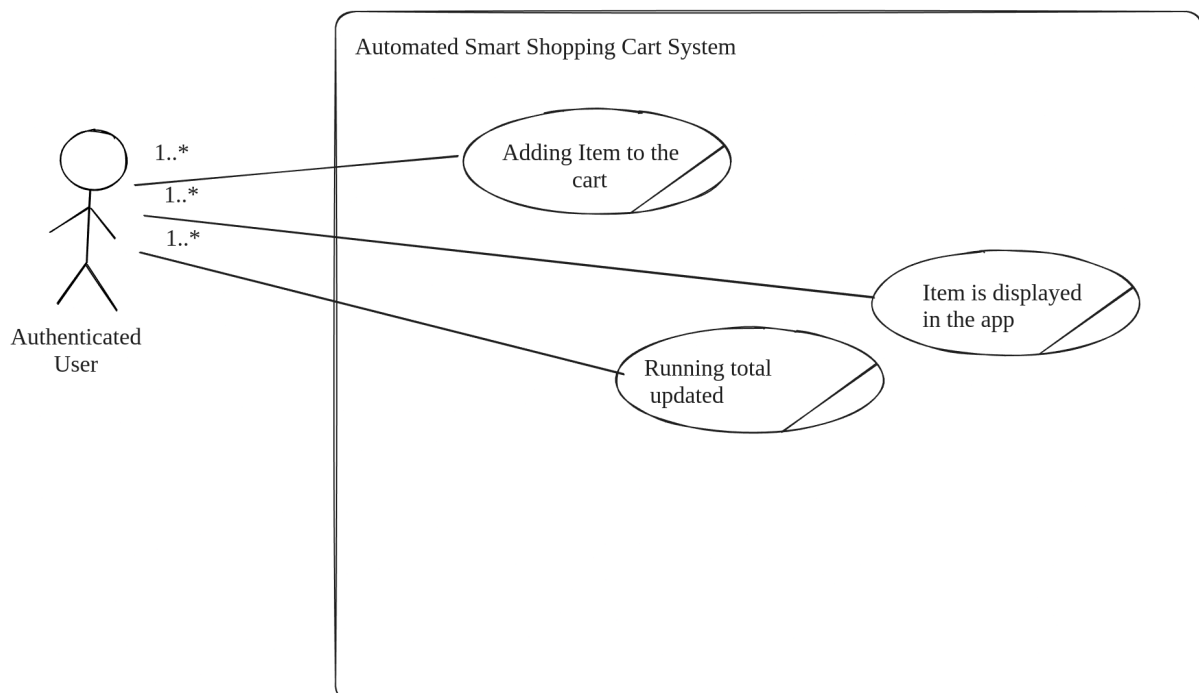


Fig. 3 - Add an item to cart Use Case

Use Case 2: User completes the checkout process

Actor: User

Preconditions: User has items in the cart and is ready to checkout.

Scenario:

1. User exits the designated store exit zone with the cart.
2. System detects cart leaving and sends notification to the app/cart device.
3. App/cart device prompts the customer for purchase confirmation.
4. Customer confirms purchase and selects payment method.
5. App/cart device sends confirmation and payment information to the system.
6. System processes payment and transmits confirmation to the store.
7. Store acknowledges confirmation and authorizes exit.

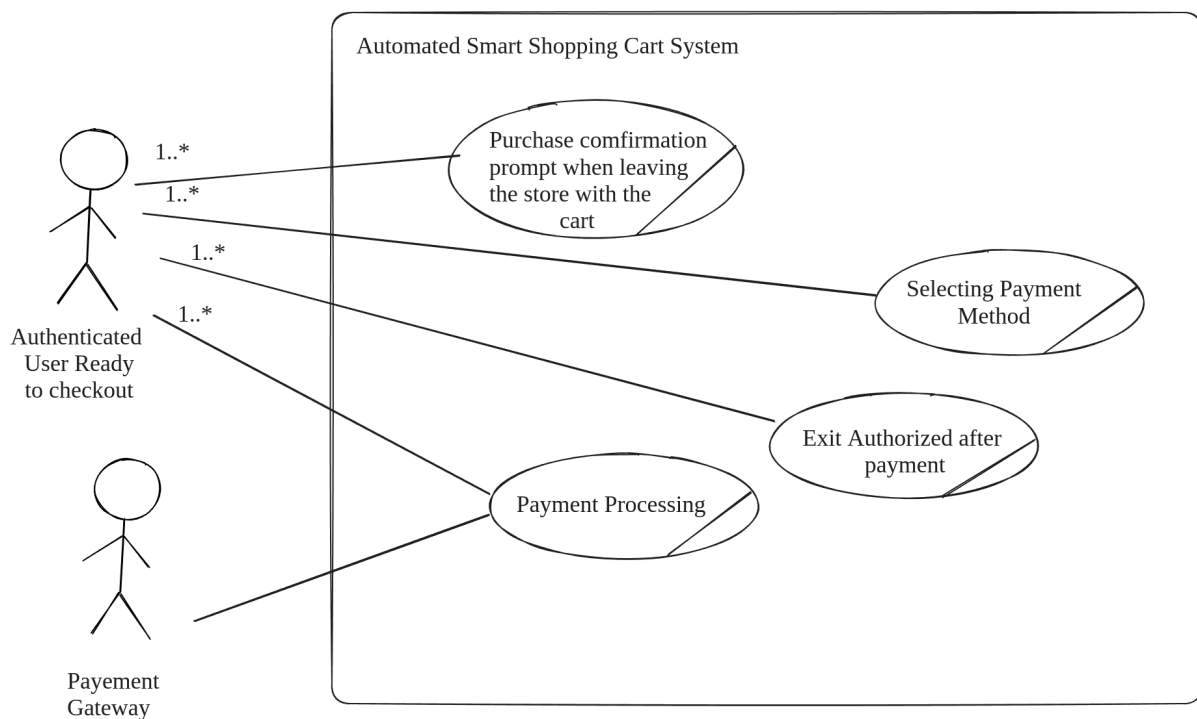


Fig. 4 - Checkout process Use Case

5. Additional Considerations

Quality Attributes

Quality Attribute	What does it mean?	Acceptance level?	How important?
Reliability	Reliability refers to the consistency and accuracy of system functionality.	Minimal downtime and accurate transaction processing are essential for ensuring reliability, meeting user expectations for availability and performance.	High. Reliability is critical for maintaining trust and credibility with users, as any downtime or errors can lead to inconvenience and damage the company's reputation.
Security	Security refers to the measures to protect the confidentiality, integrity, and availability of data and resources within the system.	The system must adhere to standard security practices, such as encryption of sensitive data and input sanitization.	Critical. Protecting user data and financial transactions is obligatory to maintain trust and credibility with customers. Any compromise in security could result in financial losses and damage to the company's reputation.
Usability	Usability concerns the ease of use and intuitiveness of the system's user interface, ensuring that users can navigate and perform tasks efficiently.	The system's user interface should undergo usability testing with a sample of users, achieving general levels of satisfaction.	High. Usability directly impacts user satisfaction and retention, influencing overall efficiency and customer experience.
Performance	Performance refers	Maintaining a short	High. A fast and

	to the responsiveness and efficiency in processing user requests.	maximum response time for user actions ensures that users experience minimal delays when interacting with the system, meeting user expectations for quick interactions.	responsive system improves user experience, increases user engagement, and contributes to higher conversion rates, driving business success and customer satisfaction.
--	---	---	--

Fig. 5 - Quality Attributes

Architectural Patterns

- **Client-Server Architecture:** The system follows a client-server model, where the client-side application interacts with server-side components for functionality such as adding items to the cart and processing payments.
- **Microservices Architecture:** The backend system is designed using microservices, allowing independent scaling and updates of functional components like user authentication and payment processing.
- **Event-Driven Architecture:** The system uses events to trigger actions, enabling real-time updates and responses to user interactions.

Architectural Decisions

- **Mobile App vs. Cart Device:** Offering both a mobile app and a cart device provides flexibility for users with and without smartphones.
- **Payment Gateway Integration:** Integration with a payment gateway facilitates secure and convenient payment processing.
- **Inventory Management System Integration:** Integrating with an inventory management system ensures accurate stock information.
- **Security Measures:** Encryption, user authentication, and secure communication protocols protect user data and transactions.
- **Usability Focus:** The user interface prioritizes ease of use and intuitiveness for an enhanced shopping experience.