HW 08 – Automated Smart Shopping Cart

ASSO 2023/2024

T22

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

Introduction

The Automated Smart Shopping Cart is designed to make shopping as smooth and enjoyable as possible, merging traditional in-store experiences with the convenience of modern technology. Taking this into consideration, we designed a system that not only simplifies the shopping process but also tackles common annoyances like long checkout lines and the hassle of manual item scanning. To better understand the architecture of this system, we created various types of structural diagrams, which showcase the system and its functionalities in different perspectives. We also created use cases, which show off the system in different scenarios and listed its quality attributes.

Structural Diagrams

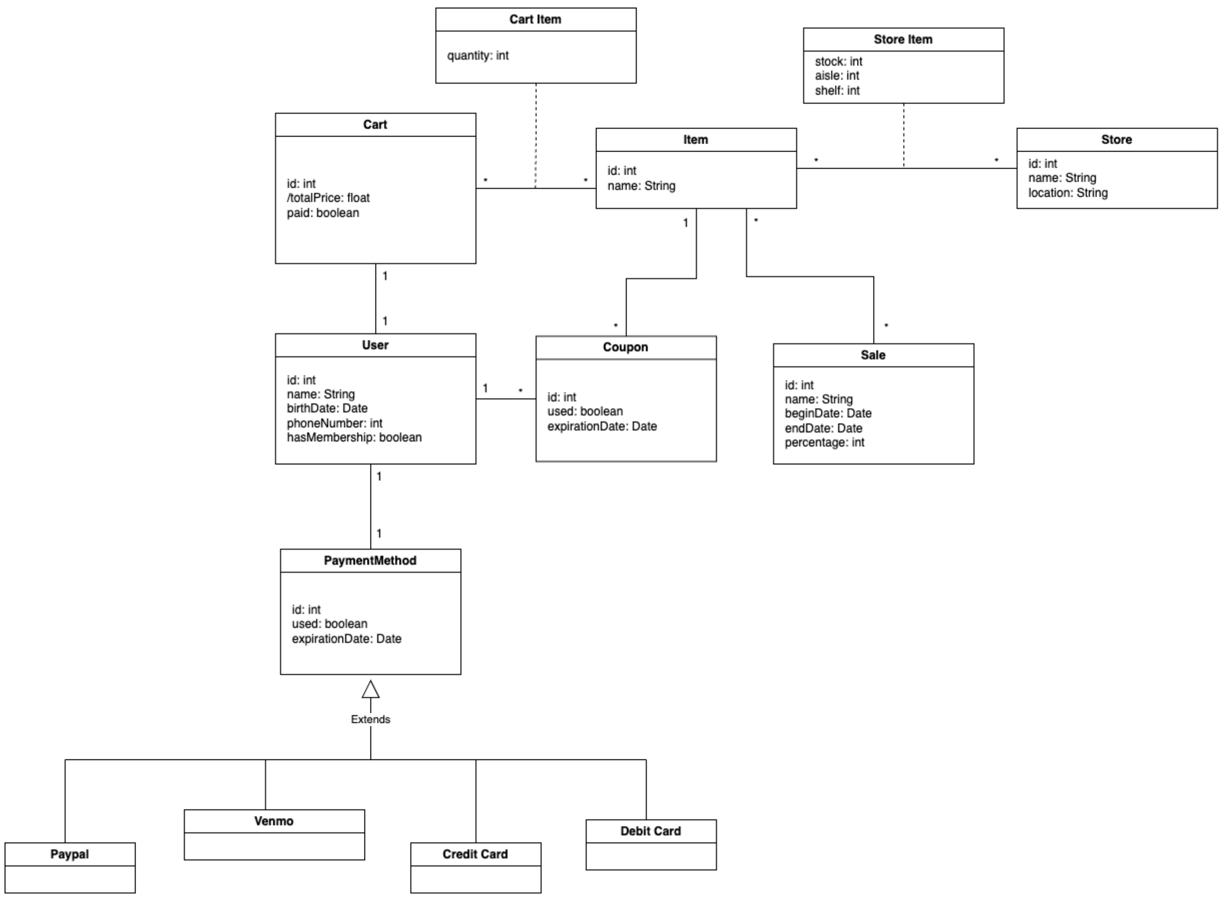

Fig. 1 – Automated Smart Shopping Cart UML Diagram

Considering the specifications given for this application, we've designed an UML diagram which has the following classes:

- **User:** Stores personal information about a customer, like their name, birthday, email, password, phone number and if they are a member or not. This information relates to a User's personal account on the mobile app.
- **Item**: Basic Information about an Item that's up for sale, like its name

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

- **Store:** Stores information about a Store, like its name and location
- **Cart**: Information about a Shopping Cart like the total price of the items in the cart, obtained by adding the price of each item, and if the cart was paid or not
- **Coupon:** Stores information about a Coupon like if it was already used and its expiration date
- **Sale:** Stores information about a Sale that happening in the stores. The information stored correspond to the name of the Sale, the begin and end dates and the discount percentage
- **PaymentMethod:** Stores information about the payment methods available. More specifically, it stores its expiration date and if it's the User's used/favorite payment method
- **Paypal, Venmo, Credit Card and Debit Card:** Specific payment methods extending the general PaymentMethod class

The diagram also showcases the following relationships:

- **One Cart can have multiple Cart Items**
- **One User is associated with one Cart**
- **One User has a favorite PaymentMethod**
- **A Store can have many Items and those Items can be sold at many stores**
- **An User can have multiple Coupons**
- **An Item can have one or more Coupons associated with it**
- **A sale can include multiple items**

It's also worth noting that, on the one hand, there can be multiples of the same item in a cart (**Cart Item**), and on the other, each store has a certain stock of each item, as well as a distinct aisle and row for it (**Store Item**).

Component View

Understanding how various functionalities work together is crucial for system design. With this in mind, the following diagram depicts the main components and interactions within this system.

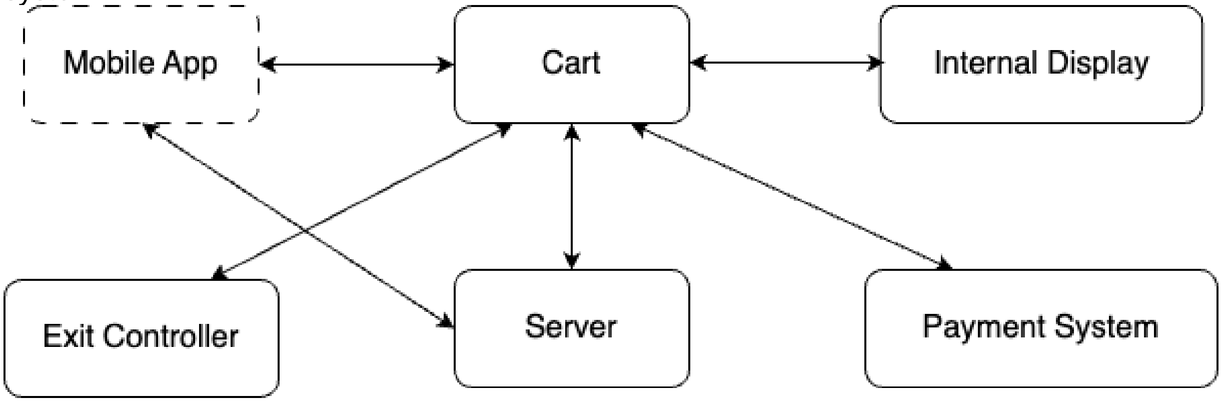

Fig. 4 – Automated Smart Shopping Cart Component Diagram

The main components of this diagram are:

- Cart: Detects items, communicates with the mobile app/cart's internal display, processes running total and facilitates payment
- Internal display: Displays the current total and the list of items

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

- Server: Processes requests and includes the store's database, which stores item and customer information, for example
- Payment System: Handles payment transactions
- Exit Controller: Identifies whenever a cart is exiting the store
- Mobile App (Optional): Can handle, for example, item scanning, show running total and item list

In this diagram, the components are associated with each other through distinct interactions:
- Mobile App-Cart: When an item is placed/removed from the cart, the item list and running total on the app are updated.
- Cart-Internal Display: If the customer doesn't have the app on them, then the cart communicates the item and total information to the internal display
- Mobile App-Server: Whenever a User searches an item by its name, its retrieves its information from the Server's database
- Cart-Server: After the checkout, the Cart sends the final bill to the Server for payment processing
- Cart-Payment System: The Cart securely transfers payment details to the payment system
- Cart-Exit Controller: Whenever the User exits the store with its cart, the Cart alerts the Exit Controller. If the User leaves without paying, the cart, through the app or its internal display, makes the alarm system go off.

Process View

With the main objective of showcasing the interaction between the parts of the system, we've made two sequence diagrams that showcase the flow of the two given key use cases:
- Adding an item to the cart (diagram #1)
- Walking out of the store (diagram #2)

Diagram #1

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

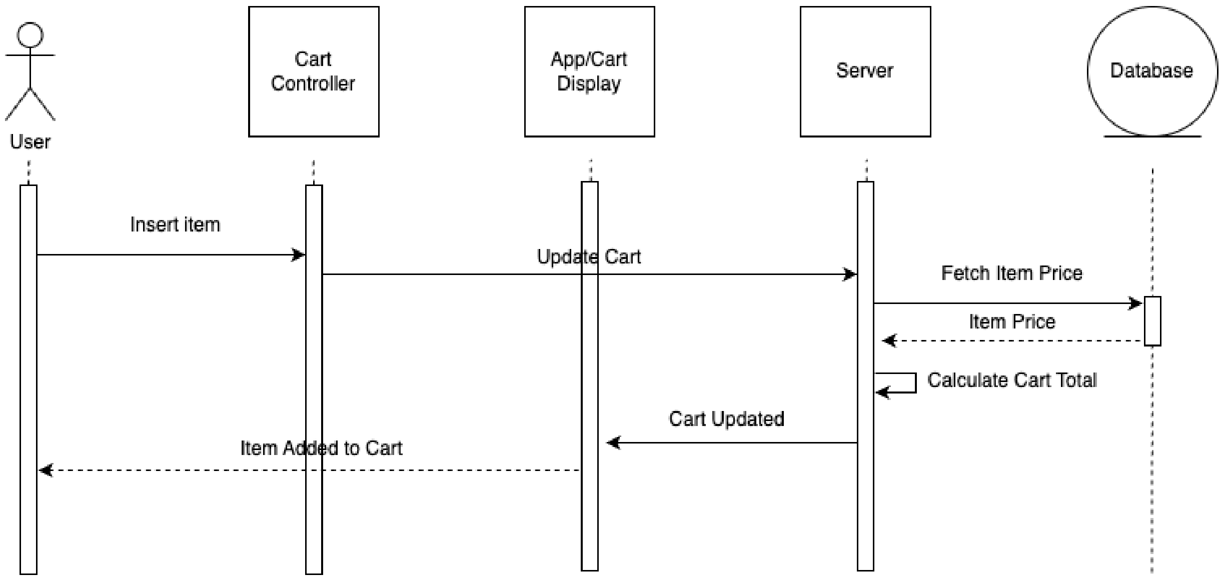

Fig. 2 – Automated Smart Shopping Cart Sequence Diagram #1

As it was said before, this sequence diagram visualizes the flow of adding an item to the cart. More specifically, it showcases the interaction between an user, the cart, the app, the server and the database. The diagram is made up of the following steps (presented chronologically):

1. **User Interaction:** The process begins with the User inserting an item in the cart.
2. **Object Detection and Identification:** The cart sensors will detect a new item and trigger the digital cart update actions by sending a server request containing the cart id, the user purchasing and the added item.
3. **Get Item Price:** The server will access the database to fetch the item's current price considering possible sales and coupons.
4. **Update Cart:** Once the item price is obtained the cart total is calculated again and the item is added to the item list. When this is done, it sends the updated cart back to the App.
5. **User Confirmation:** The user receives visual confirmation on the app and/or cart display that the item has been added.

Diagram #2

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

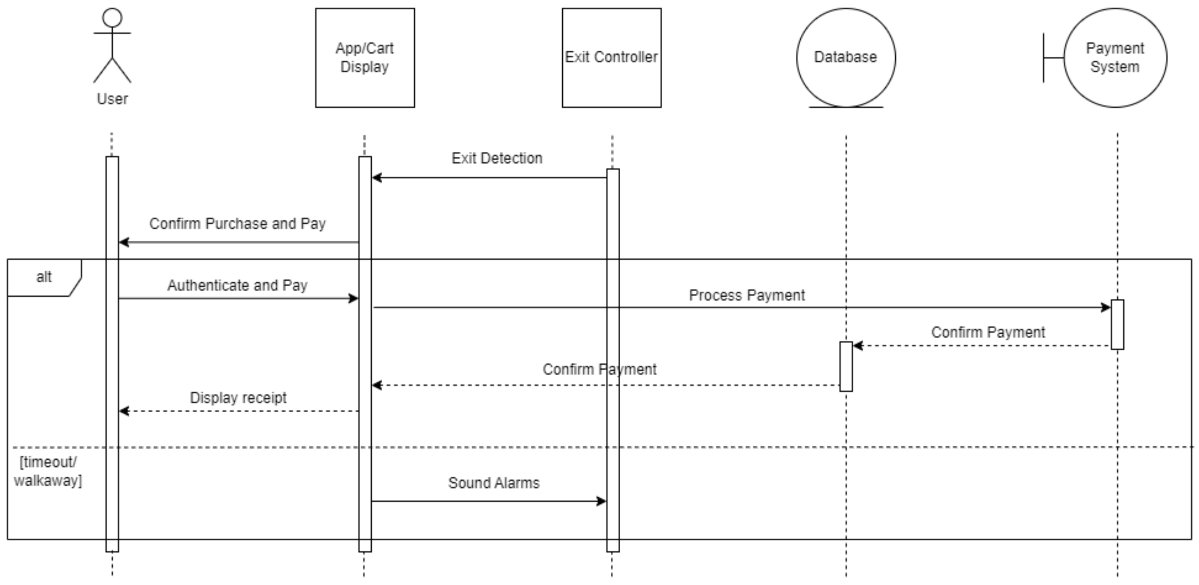

Fig. 3 – Automated Smart Shopping Cart Sequence Diagram #2

Regarding this sequence diagram, its main objective is to highlight what happens when a user decides to leave the store. The diagram goes as follows:

1. **Exit Detection**: The user walks towards the store exit with the smart cart. Exit sensors detect the cart and signal the cart's controller.
2. **Payment Request**: The cart prompts the user to confirm the purchase on the app and/or cart display.
3. **Payment Confirmation**: The user undergoes security authentication (e.g., PIN entry).
4. **Payment Processing**: Upon successful authentication, the payment is processed through the favorite method (credit/debit card, Venmo, etc.).
5. **User Confirmation**: A digital receipt is generated and displayed/sent to the user.
6. **Alarm Deactivation**: The store's security system deactivates the alarm for this cart.

If step 3 fails, either because the user simply ignores the request for too long or if it is detected that the user is walking away without paying, the app will signal the alarms to turn on.

Use Case View

The outlined use cases here describe how the users (shoppers) interact with the smart cart and the technology that supports it, from picking up an item and putting it in the cart to the final steps of checking out.

The first two use cases below demonstrate key functionalities, interactions, and their critical role in maintaining an efficient and user-friendly shopping experience. We decided to explore the first two in detail, but we also listed seven other use cases that help paint a picture of how the smart cart system facilitates the transition between choosing products and making a purchase.

Use Case 1: Adding an Item to the Cart

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

Title: User adds item to cart

Description: This use case details the process by which a user adds an item to their smart shopping cart and the system updates accordingly.

Actors: User and Smart Cart

Preconditions:

- The user must have the smart cart app installed on their phone and/or use the cart's integrated display.
- The cart is operational and connected to the store network.

Postconditions:

- The cart's item list and running total are updated.
- Any applicable discounts or coupons are applied to the running total.

Main Path:

1. The user selects an item and places it in the smart cart.
2. The item sensor detects the item entering and identifies it
3. The smart cart sends item data to the server to fetch price and discount information.
4. The server calculates the total cost considering any active discounts or coupons and sends it back to the cart.
5. The smart cart updates the running total and item list on the user’s app and/or the cart's display.
6. The user receives visual confirmation on the app and/or cart display that the item has been added.

Alternate Path: If the item cannot be identified, the user will need to manually enter the item’s barcode or ask for assistance.

Exception Path: If the network connection is lost during the addition, the cart stores the data locally and attempts to update once the connection is restored.

Priority: High -> This use case is essential for the core functionality of the architecture.

Frequency of Use: High -> This action will be performed multiple times during a typical shopping session.

Use Case 2: Checking out of the Store

Title: Cart automatic checkout upon exiting

Description: This use case outlines the process that occurs when a user exits the store with a smart cart, triggering automatic checkout.

Actors: User, Smart Cart, Payment System and Store Security System

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

Preconditions:

- The user has finished shopping and the cart contains at least one item.
- The cart is operational and connected to the store network.

Postconditions:

- The purchase is confirmed and processed.
- The receipt is generated and provided to the user.
- Store security system is informed of the legitimate exit.

Main Path:

1. The user walks towards the store exit with the smart cart.
2. Exit sensors detect the cart and signal the cart's controller.
3. The smart cart prompts the user to confirm the purchase on the app and/or cart display.
4. The user undergoes security authentication (e.g.: PIN entry).
5. Upon successful authentication, the payment is processed through the favorite method (e.g.: Credit/debit card, Venmo, etc.).
6. A digital receipt is generated and displayed/sent to the user.
7. The store's security system deactivates the alarm for this cart.
8. The user leaves the store with the smart cart.

Alternate Path: If the user opts to add more items, they can return to the shopping area, and the purchase won’t happen.

Exception Path: If the user fails to confirm the purchase, the store alarm activates.

Priority: High - This use case is essential for the core functionality of the architecture.

Frequency of Use: High - Every shopping trip concludes with this process.

Note: Priority and Frequency of Use were classified in a five-tier classification: Low, Low to Medium, Medium, Medium to High and High.

List of additional use cases

1. User logs In/Out of the Smart Cart App, during shopping
2. User checks item details
3. User requests assistance
4. Cart applies user coupons to the purchase
5. Cart abandonment
6. Cart alert for Battery Low
7. Cart synchronization with User Account

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

Use Case Map

T22
António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

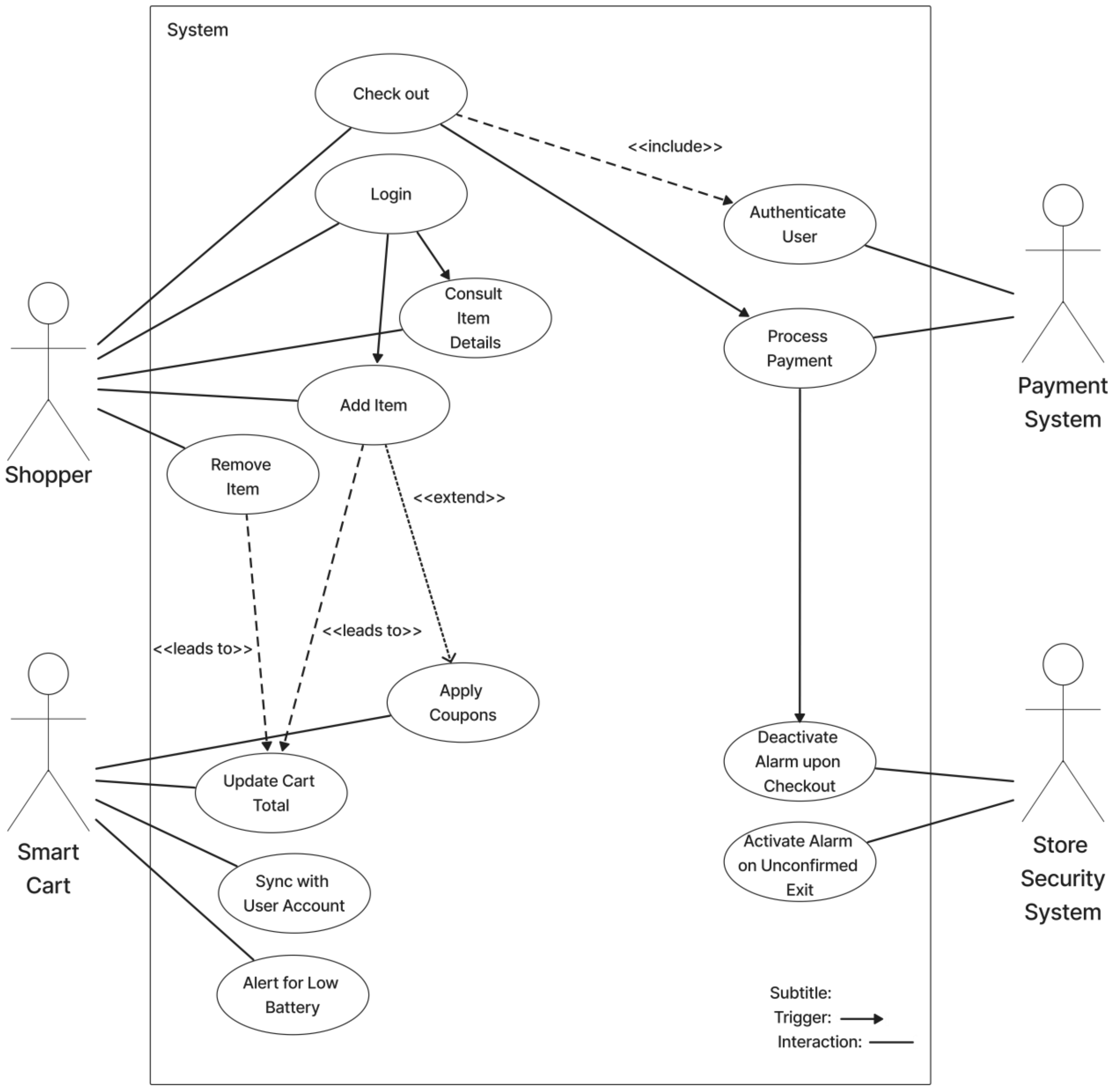

Fig. 5 – Smart Cart Architecture's Use Case Map

Quality Attribute Requirements

Quality Attribute	What does it mean?	Acceptable level?	How Important?

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

Security	Ability to safeguard critical information and data from unauthorized access.	All data transactions are encrypted, and strict authentication measures (with a PIN or other security measure) are implemented for user login and payment processes.	5 The system needs to always prevent data breaches and protect user privacy, which is critical for maintaining trust with the users and compliance with regulations.
Usability	Related to ease of use; The system should be intuitive and straightforward, guiding users through the process of adding and removing items and checking out.	Users can navigate the shopping process and checkout with minimal assistance.	4 If the system is difficult to use, it could lead to frustration, reducing customer satisfaction and potentially driving customers away.
Portability	The system's ability to function across different devices and platforms, particularly mobile devices used in the store.	The system is compatible with major mobile operating systems (iOS, and Android) and the cart’s integrated hardware interfaces smoothly with the app.	3 The system ensures accessibility and convenience for a wide range of users, enhancing user engagement across different devices. It gives the user the choice to use the cart’s display or their mobile phone, using the App.
Scalability	Refers to the ability to handle increases in users, carts, transactions, and data load without performance degradation.	The system maintains responsive times even during peak shopping hours and can scale during high traffic events like holiday sales (like Black Friday)	4 The system must handle high loads, especially during peak times, to ensure continuous service without delays or crashes.
Maintainability	Ensures the system can be easily updated, modified, and corrected post-deployment.	The system supports straightforward implementation of updates and new features, quick bug correction, and performance	4 The system should be easy to update and maintain, particularly when adding new features or during

António Ferreira – up202004735
Francisco Serra – up202007723
Gonçalo Almeida – up202308629
João Maldonado – up202004244
Tomás Gomes – up202004393

		enhancements without bigger downtime.	routine database updates.

Table 1 - Smart Cart Architecture's Quality Attributes

“How Important” Scale: 1 - Not important, 2 - Slightly important, 3 - Moderately Important, 4 - Very Important, 5 - Extremely Important