ASSO - *Automated Smart Shopping Cart*

Homework 8

Team 33
Guilherme de Matos Ferreira de Almeida
João Pedro Carvalho Moreira
Jorge Daniel de Almeida Sousa
Lia da Silva Linhares Vieira
Nuno Afonso Anjos Pereira

U.PORTO

Masters in Informatics and Computing Engineering

17/04/2024

Contents

1 Introduction

Online shopping has spoiled us with its convenience, but physical shopping lacks the same efficiency. To bridge this gap, we present an automated smart shopping cart that simplifies the in-store shopping experience. This system automatically detects when items are added to or removed from the cart, maintaining a continuously updated total. You can easily view the current contents of your cart and the running total at any time. When you're ready to leave, simply confirm the charges to your payment method, and you're good to go without any waiting at the checkout counter.

In this paper, we delve into the architecture and design considerations of our automated smart shopping cart system, highlighting the key components and their relationships. Additionally, we explore how these components connect to facilitate seamless interactions within the system and outline the quality attribute requirements, ensuring that the system meets essential criteria such as usability, reliability, security, and scalability.

2 Structure Diagrams

2.1 Class Diagram

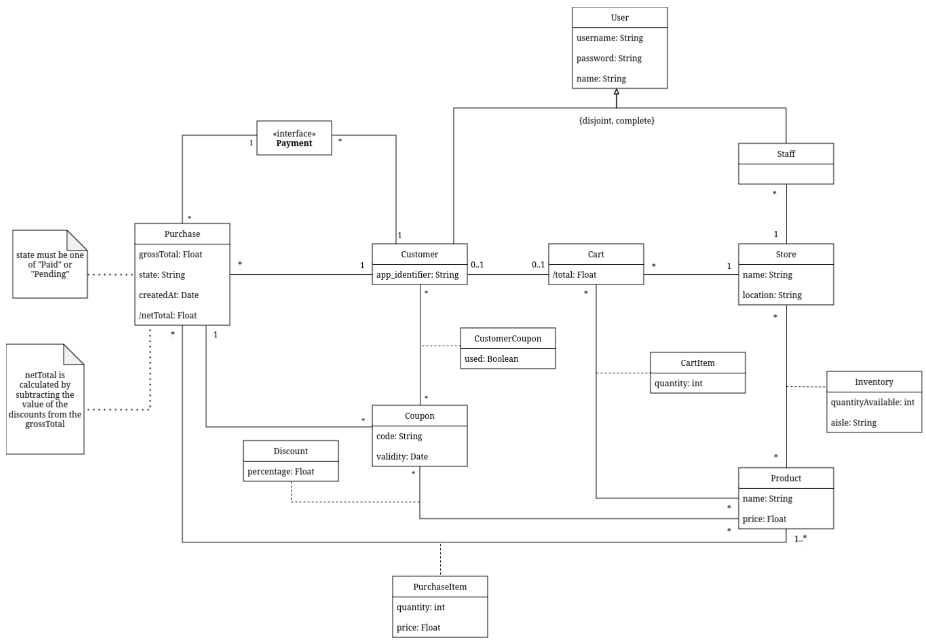

Figure 1: Class Diagram

2.1.1 Store

Represents a physical establishment where products are sold. It includes information such as the store's name and location.

2.1.2 Product

Represents an item available for purchase in the store. It includes information such as the product's name, price, and aisle location within the store.

2.1.3 Cart

Represents a physical shopping cart. It keeps track of the running total.

2.1.4 Coupon

Represents a promotional offer that provides customers with discounts on specific products or purchases. It includes information such as the coupon code and the expiration date.

2.1.5 Purchase

Represents a purchase transaction made by a user for products. It includes information such as the current state of the purchase (which can be *paid* or *pending*), the date, the gross total, and the net total. The net total is a derived attribute calculated by subtracting the value of discounts from the gross total.

2.1.6 User

Represents individuals who interact with the system. It includes basic user information such as name, username, and password. Users can be further categorized into Customers or Staff.

2.1.7 Customer

Represents a specialized type of user within the system. Customers are individuals who engage in purchasing products within the store.

2.1.8 Staff

Represents a specialized type of user within the system. Staff members are employees or administrators who have access to system functionalities such as setting prices, managing inventory, and applying discounts. Staff accounts are authenticated within the system to ensure secure access to sensitive functionalities.

2.1.9 Payment (Interface)

Represents the payment interface utilized for processing payments. This interface allows the system to integrate with various third-party payment service providers. Implementations of this interface handle payment authorization, transaction processing, and communication with external payment gateways, providing flexibility and security in payment processing operations.

2.2 Architectural Overview

An architectural diagram representing our proposed solution is as follows:

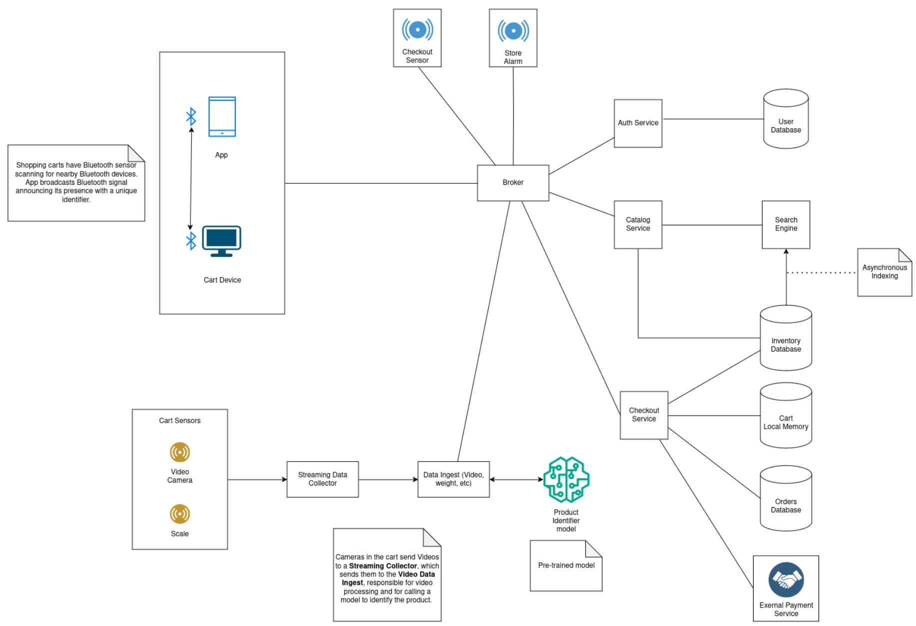

Figure 2: Architectural Overview

The architectural diagram in Figure 2 illustrates the various components and their interactions within the automated smart shopping cart system.

Shopping carts are equipped with **Bluetooth sensors** scanning for nearby Bluetooth devices. The accompanying **mobile app** broadcasts a Bluetooth signal announcing its presence with a unique identifier. A **broker** facilitates communication between the app, checkout sensor, store alarm, and various services including Catalog, Authentication, and Checkout Services.

Cameras installed in the cart capture videos of the cart's contents, which are sent to a **Streaming Collector**. The Streaming Collector forwards the videos to the **Video Data Ingest** service responsible for video processing. The Data Ingest service processes the videos and calls a **pre-trained model** to identify the products. Each cart is equipped with a **Local Memory Store** to store the products it is currently holding at any given time. This removes some of the communication that would happen between the system's components should the cart's contents be stored in a centralized database, since product addition and removal is a frequent operation: this way, the system is not flooded with requests and can handle higher loads of other types of requests.

The **Checkout Service** connects directly to the **Inventory Database**, Cart Local Memory, Orders Database, and Payment Service, facilitating seamless checkout processes.

A **Search Engine** retrieves data from the Inventory Database, which is utilized by the **Catalog Service** to provide product search capabilities to users.

The **Authentication Service** manages user authentication and authorization, utilizing a User Database to authenticate users accessing the system. Using a separate service and data store for user authentication, authorization, and storage allows the system's needs to change and grow independently of who the users are or how they are stored. This also opens the door to utilizing external services, which can bring long-term benefits in the form of no risk of failing to comply with local and government regulations regarding data privacy and data security.

Overall, the architectural diagram illustrates the "interconnectedness" of various components within the system, facilitating efficient product detection, checkout processes, and user authentication.

Some key decisions had to be made regarding the architecture and its implications, particularly concerning product detection inside a shopping cart, described in section 2.2.1.

2.2.1 Product detection inside a shopping cart

For the detection of products inside the shopping cart, including the addition or removal of new items, we considered 2 solutions:

- Computer Vision;[1]
- RFID tags;[2]

Additionally, the integration of **weight-based product detection** could enhance the accuracy and security of the system when used in conjunction with these solutions.

Both solutions have their pros and cons, which are explored further.

2.2.1.1 Computer Vision

Deploying a **Computer Vision**-based solution would require the addition of several cameras collecting input from multiple angles of the shopping cart. This would allow the system to have an unblocked view of the cart's contents at any time.

During normal operation, the camera sensors would collect a video feed and send it to a **Video Data Ingest** module/service, responsible for collecting, processing, and analyzing this data. Alongside an **image recognition model**, this service would be responsible for identifying the products that are inserted and removed into the cart.

Upon identifying a given product, it sends a message to the **Product Cart Service**, which then correctly updates that customer's shopping cart information.

2.2.1.2 RFID

Using an **RFID**-based approach implies that not only the shopping cart but also the individual products are equipped with some form of RFID tagging.

The cart's RFID sensor would continuously scan its interior for any products it might find. This way we don't need to account for both insertions and removals, only presence is enough.

The cart would also be equipped with a local data store to register what products were inside it at any given time. This helps in the checkout process since each cart only knows - and only needs to know - about its own products. It also removes some dependencies with the store's internal system, allowing for more decentralized operations.

2.2.1.3 Scale-based Detection

Integrating scales into the shopping cart offers an additional layer of assurance for product detection. By measuring the weight of items added or removed from the cart, the system can detect discrepancies between expected and actual weights, indicating potential issues such as unscanned items or attempted theft. When used alongside Computer Vision and RFID solutions, scale-based detection enhances accuracy and security, providing a comprehensive approach to product detection in shopping carts.

2.2.1.4 Comparison

On one hand, using Computer-Vision technologies would require more equipment per cart since only one sensor is not sufficient for effectively tracking the cart's contents. Since the product identification is handled by an image recognition model, no further hardware is needed, relieving some of the cost from the products themselves. However, using this approach requires a separate image processing module whose task would only be to process images and flag any products found in them, which is a computationally expensive task.

On the other hand, using RFID would require the purchase of RFID sensors for every cart and RFID tags for every product that is available to consumers in the store, potentially increasing costs significantly. RFID sensors and tags vendor prices would need to be accounted for when deciding the best solution for this problem. Another potential issue that might arise from the usage of RFID sensors and tags is if the cart is too close to products in the display: there can be a false-positive in which a product is flagged as being in the cart while in reality is not. One way to prevent this is to electrically and magnetically isolate the interior of the shopping cart, so the cart's RFID sensor can only "see" what is inside of it. Despite all this, opting for this solution would require less equipment per cart and less computing power overall, since there is no need for a separate module responsible for processing the data, nor to deploy an entire image recognition model. Upon checkout, we can reuse the systems already in place to process product purchases.

2.2.1.5 Conclusion

Despite its pros and cons, we believe the best option for product detection inside a shopping cart would be **Computer Vision** due to its ability to provide real-time visual recognition of products without the need for additional hardware on each item.

While RFID offers a streamlined approach, the cost associated with RFID tags and sensors could outweigh its benefits in certain scenarios. Furthermore, since this product is expected to be sold to various supermarkets, they would need to install RFID on all their products before using this shopping cart solution. In our opinion, this makes the product less appealing, while using computer vision makes it more *plug-and-play*, without many downsides associated.

Additionally, integrating **scale-based detection** enhances the accuracy and security of the system when combined with Computer Vision, providing a comprehensive solution for product detection in shopping carts.

3 Use Cases

The following is a list of the most expected use cases of the systems, for which our architecture was thought out:

1. **Add Item to Cart:** A customer places an item into the shopping cart.
2. **Remove Item from Cart:** A customer removes an item from the shopping cart.
3. **View Cart Contents and Total:** A customer checks the list of items in the cart and the running total.
4. **Confirm Purchase:** A customer selects a payment method and enters the PIN to confirm their purchase.
5. **Locate Product:** A customer searches for the location of a product.
6. **Report Feedback:** A customer provides feedback on their shopping experience, including satisfaction levels, suggestions, or complaints.
7. **Manage Payment Methods:** A customer adds or updates their payment methods.
8. **Customer Identification:** The system links the customer's phone to the cart and detects the customer's identity.
9. **Apply Coupons and Discounts:** The system applies coupons and discounts to the purchase.
10. **Update Product Information:** A staff member updates the price, discounts, or inventory of the products.

4 Sequence Diagrams

The following sequence diagrams describe the sequences of messages performed throughout the entire system for what we believe are the 2 most important use cases of the system: **product addition** and **automatic checkout**.

4.1 Use Case 1: Add item to cart

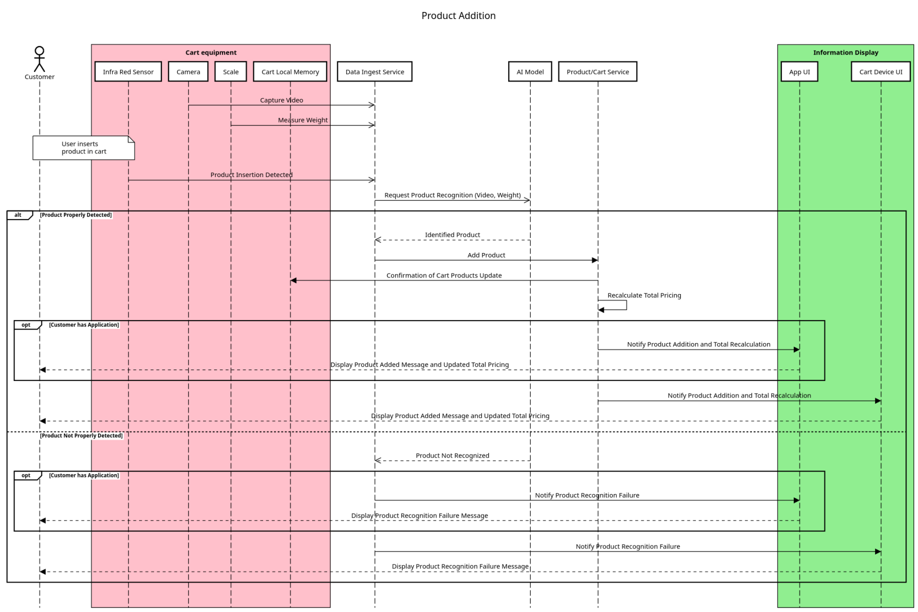

Figure 3: Product Addition Sequence Diagram

The sequence diagram in Figure 3 illustrates the process of adding an item to the cart within the system. The process starts with the Infra-Red Sensor detecting the insertion of a product into the cart. Then, the Data Ingest Service coordinates with the Camera and Scale to capture video and measure weight, respectively. The AI Model is then requested to recognize the product based on the captured data. If the product is properly detected, it is added to the cart by the Product/Cart Service, and the total pricing is recalculated. Notifications are sent to both the App UI and Cart Device UI to inform the customer about the addition and update of total pricing. If the product is not properly detected, appropriate failure notifications are displayed to the customer.

4.2 Use Case 2: Checkout

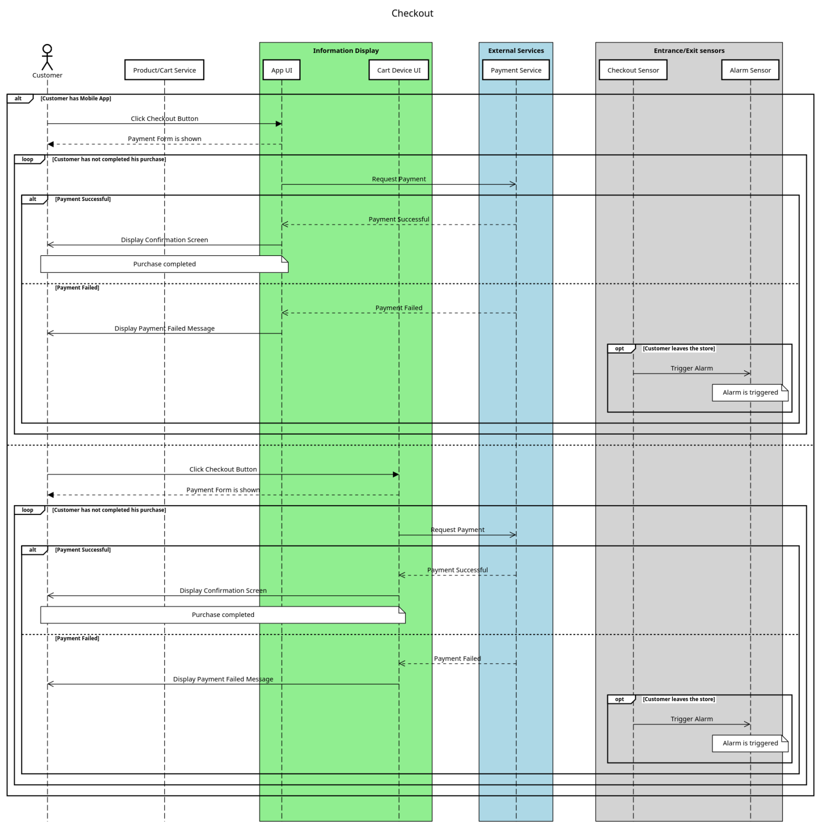

Figure 4: Checkout Sequence Diagram

The sequence diagram in Figure 4 depicts the automatic checkout process within the system. The flow varies depending on whether the customer has a mobile app or not. If the customer has a mobile app, they can choose to pay or not pay after clicking the checkout button. If the payment is successful, confirmation screens are displayed, and the purchase is confirmed. If the payment fails or the customer leaves without paying, the store alarms are triggered. If the customer doesn't have a mobile app, similar steps occur through the Cart Device UI instead.

5 Quality Attribute Requirements

As with anything, many quality attributes had to be taken into consideration when designing and developing the system. Some of the more relevant attributes for our specific problem are:

5.1 Usability

The usability of the system is extremely important to ensure a positive shopping experience for customers. The system must have a user-friendly interface that allows for seamless interaction both in-store and via the mobile app. By prioritizing usability, we aim to reduce user frustration and encourage adoption of our system.

5.2 Reliability

Customers rely on the automatic checkout system to accurately detect items added or removed from the cart and provide reliable transaction processing. Reliability is fundamental to the success of the automatic checkout system. Ensuring robust reliability measures safeguards against errors, instilling confidence in the system's functionality and building user trust.

5.3 Availability

The system's availability is crucial to meet the needs of customers at any time, whether during store hours or beyond. It should be accessible whenever customers require its services, including tasks like confirming purchases outside of normal operating times. By maintaining high availability, we ensure continuous service delivery, meeting customer demands and expectations effectively.

5.4 Security

Security is of great importance in protecting customer information and preventing unauthorized access or transactions. Implementing secure authentication methods, such as PIN entry, adds an extra layer of protection to confirm purchases securely. By prioritizing security measures, we safeguard customer data and financial transactions, fostering trust and confidence in the system's integrity.

5.5 Scalability

Scalability is essential to accommodate fluctuating numbers of users and transactions, especially during peak shopping hours. The system's architecture must be designed to seamlessly scale up or down in response to changing demands, ensuring optimal performance and resource utilization. By prioritizing scalability, we guarantee that the system can effectively handle increased workload without sacrificing performance or reliability.

5.6 Performance

Performance plays a crucial role in delivering real-time updates and maintaining minimal latency in the system. Quick detection of items and instantaneous updates on the running total and cart contents are imperative to enhance the shopping experience. By optimizing performance, we ensure swift and responsive interactions, meeting customer expectations and maximizing user satisfaction.

5.7 Interoperability

Interoperability is essential for seamless integration with the store's existing infrastructure, including inventory management systems and payment gateways. Compatibility and smooth communication between different components of the system are vital to ensure cohesive operation and data exchange. By prioritizing interoperability, we facilitate efficient collaboration and streamline processes, enhancing the overall functionality and effectiveness of the system.

6 Conclusion

In conclusion, the automated smart shopping cart system, as detailed in the report, presents a thorough architectural design aimed at enhancing the traditional in-store shopping experience. Through the integration of technologies such as computer vision or RFID, the system offers functionalities like real-time item detection and seamless checkout processes. Quality attributes such as usability, reliability, security, scalability, and interoperability are meticulously considered in the design, ensuring a robust and efficient system. While the system shows promise in streamlining operations for retailers and improving convenience for customers, further evaluation and testing may be necessary to validate its effectiveness and address any potential limitations.

References

[1] Wikipedia contributors. Computer vision — Wikipedia, the free encyclopedia, 2024. [Online; accessed 21-April-2024].

[2] Wikipedia contributors. Radio-frequency identification — Wikipedia, the free encyclopedia, 2024. [Online; accessed 21-April-2024].