Faculty of Engineering of the University of Porto

Homework 08

Automated Smart Shopping Cart

M.EIC010 - Software Systems Architecture

1MEIC03 - T32

Professors

Ademar Aguiar

Neil Harrison

Students

Ana Rita Baptista de Oliveira - up202004155@edu.fe.up.pt

Diogo Alexandre da Costa Melo Moreira da Fonte - up202004175@edu.fe.up.pt

João Paulo Moreira Araújo - up2020004293@edu.fe.up.pt

Tiago Nunes Moreira Branquinho - up202005567@edu.fe.up.pt

Introduction

Nowadays, the demand for a faster and more convenient checkout experience in physical shopping is becoming increasingly similar to the ease of online shopping. Traditional shopping involves many tedious steps like waiting in line, waiting for items to be scanned, and loading everything into the cart after having to remove it for scanning. The need for a better checkout experience is becoming more and more apparent.

In this report, we will explore a possible architectural design for a smart automated shopping cart that allows customers to add, remove, and pay for items in their cart, without the need for a traditional checkout process.

Architecture Design

The smart automated shopping cart system is designed to provide a seamless experience for users looking to have a fast and easy checkout experience on their shopping. This architecture aims to ensure reliable addition and removal of items, an efficient and fast checkout process, and an overall fast in-and-out shopping experience while maintaining system scalability and performance to its core.

The following are the views requested for this report in the system.

Logical View

For the Logical View of the system, we'll start by identifying the major components and their relationships to each other:

Components

- **Cart Manager**
    - Responsible for managing cart-related operations.
    - Subcomponents:
        - **Item Manager:** Receives and handles messages from the hardware that detects items as they are added/removed from the shopping cart.
        - **Phone Detection System:** Detects the mobile phones of people with the app installed and associates the closest phone to each cart.
        - **Entry and Exit Detection System:** Detects if a cart enters or exits the store and, if the cart exits without proper payment, triggers the Security System.

        - **Checkout:** Allows users to terminate their purchase, confirming payment information and transaction. Also re-calculates prices if the authenticated customer has any applicable coupons.
- **User Manager**
    - Manages user-related operations such as authentication, user profile management
    - Subcomponents:
        - **Authentication Service:** Allows users to authenticate with an account.
        - **Profile Management:** Manages user profiles, including payment methods and preferences.
- **User Interface (UI)**
    - Allows users to interact with the automatic checkout system.
    - Provides the ability to search for the details of a certain item (aisle number, price, or other).
    - Subcomponents:
        - **Mobile Application:** Allows users to interact with the automatic checkout system through their phones.
            - **Item Catalog Interface**: Displays available items and their basic information. Has the possibility to search for specific items.
            - **Authentication Interface**: Allows users to authenticate with an account to purchase on the store, with an associated payment method.
            - **Account Management Interface**: Allows users to manage their profile information and preferences.
            - **Confirmation Interface**: Allows users to confirm the payment method and checkout, to be able to finish the purchase.
        - **Cart Application:** Allows users to interact with the automatic checkout system when they don't have a phone with the mobile app.
            - **Item Catalog Interface**: Displays available items and their basic information. Has the possibility to search for specific items.
            - **Payment Interface**: Allows users to confirm the billing address

and pay with a debit/credit card in the cart device.

- **Inventory Management System**
    - Syncs real-time inventory data with the automated checkout. Also allows store owners to update inventory.
    - Subcomponents:
        - **Item Catalog Manager:** Stores details about available items, such as name, price, stock, and aisle.
- **Security System**
    - Ensure the security of the automatic checkout process.
    - Subcomponents:
        - Alarm System: Triggers the store alarm systems for unauthorized activity (leaving the store without paying)
        - Cart Monitoring: Monitors cart activity to prevent theft or tampering.
        - Item Pick-up Monitoring: Monitors items that are picked up but not inserted into the cart, through the use of cameras and AI models to detect the items and the carts.
- **Payment Gateway**
    - Interfaces with external payment processing services to handle payment transactions securely. Includes debit/credit card, as well as other payment methods like Venmo.
- **Items Inventory Database**
    - Store information regarding items for sale, including name, price, and stock (weight or units).
- **User Database**
    - Stores information about the users, including profile, payment details, purchase history, coupons, and discounts/sales.

Connectors

To ease cluttering the diagram, you can link the connections to the respective number. Taking this information in mind, these are the components:

1. **Mobile UI Item Catalog - Item Catalog Manager:**
    - This connector facilitates communication between the User Interface of the Mobile Application and the Item Catalog Manager, allowing retrieval of information about available items for display to users.

2. **Mobile UI Account Management - Profile Management:**
    - This connector allows the user to modify or add personal information like name, contact, and address. It also allows the user to add/remove several payment methods and choose one of them to be the predefined one.
3. **Mobile UI Authentication - Authentication Service:**
    - This connector enables the User Interface to communicate with the Authentication Service component, allowing users to authenticate their accounts, to be able to check out the purchase through the mobile app.
4. **Mobile Application - Phone Detection System:**
    - This connector facilitates the communication between the mobile application and the cart of the user. This connector is responsible for the link between the user's cart and the user's application/mobile phone.
5. **Mobile Application - Cart Item Manager:**
    - This connector is responsible for synchronizing the items inserted in the cart and having an updated list of items (with respective prices) and the current total price on the Mobile Application.
6. **Mobile Application Confirmation - Checkout:**
    - This connector links the Mobile User Interface to the Checkout component, initiating the checkout process after users have confirmed the items, selected payment method, and reviewed pricing details.
7. **Cart UI Item Catalog - Item Catalog Manager:**
    - This connector facilitates communication between the User Interface of the Cart Application and the Item Catalog Manager, allowing retrieval of information about available items for display to users.
8. **Cart Application - Cart Item Manager:**
    - This connector is responsible for synchronizing the items inserted in the cart and having an updated list of items (with respective prices) and the current total price on the Cart Application.
9. **Cart Application Payment - Checkout:**
    - This connector links the Cart User Interface to the Checkout component, finishing the purchase process after users have confirmed the items and paid with a debit/credit card, using the payment terminal of the cart.
10. **Checkout - Payment Gateway:**

- This connector connects the Checkout component to the Payment Gateway, facilitating the processing of payment transactions securely, using the predefined payment method in the mobile application.

11. **Entry and Exit Detection System - Alarm System:**
    - This connector allows the Entry and Exit Detection System to interact with the Alarm System. When a cart leaves the store without proper purchase confirmation and payment, the store alarm systems need to be triggered.
12. **Entry and Exit Detection System - Cart Monitoring:**
    - This connector allows the Entry and Exit Detection System to interact with the Cart Monitoring component. When a cart enters or leaves the store, the Cart Monitoring component needs to communicate this to the Cart Manager, to decide what to do.
13. **Item Pick-up Monitoring - Cart Item Manager:**
    - This connector allows the communication between the Item Pick-up Monitoring component and the Cart Item Manager, to be able to detect when items are picked up but not inserted into the cart, to prevent theft.
14. **Authentication Service - User Database:**
    - This connector facilitates communication between the Authentication Service and the User Database component, allowing the service to verify user credentials against stored user information securely.
15. **Profile Management - User Database:**
    - This connector allows the Profile Management component to interact with the User Database, enabling the management and retrieval of user profile information such as preferences and payment methods.
16. **Item Catalog Manager - Item Inventory Database:**
    - This connector links the Item Catalog to the Item Inventory Database component, providing access to item-related data such as item details, stock availability, and pricing.

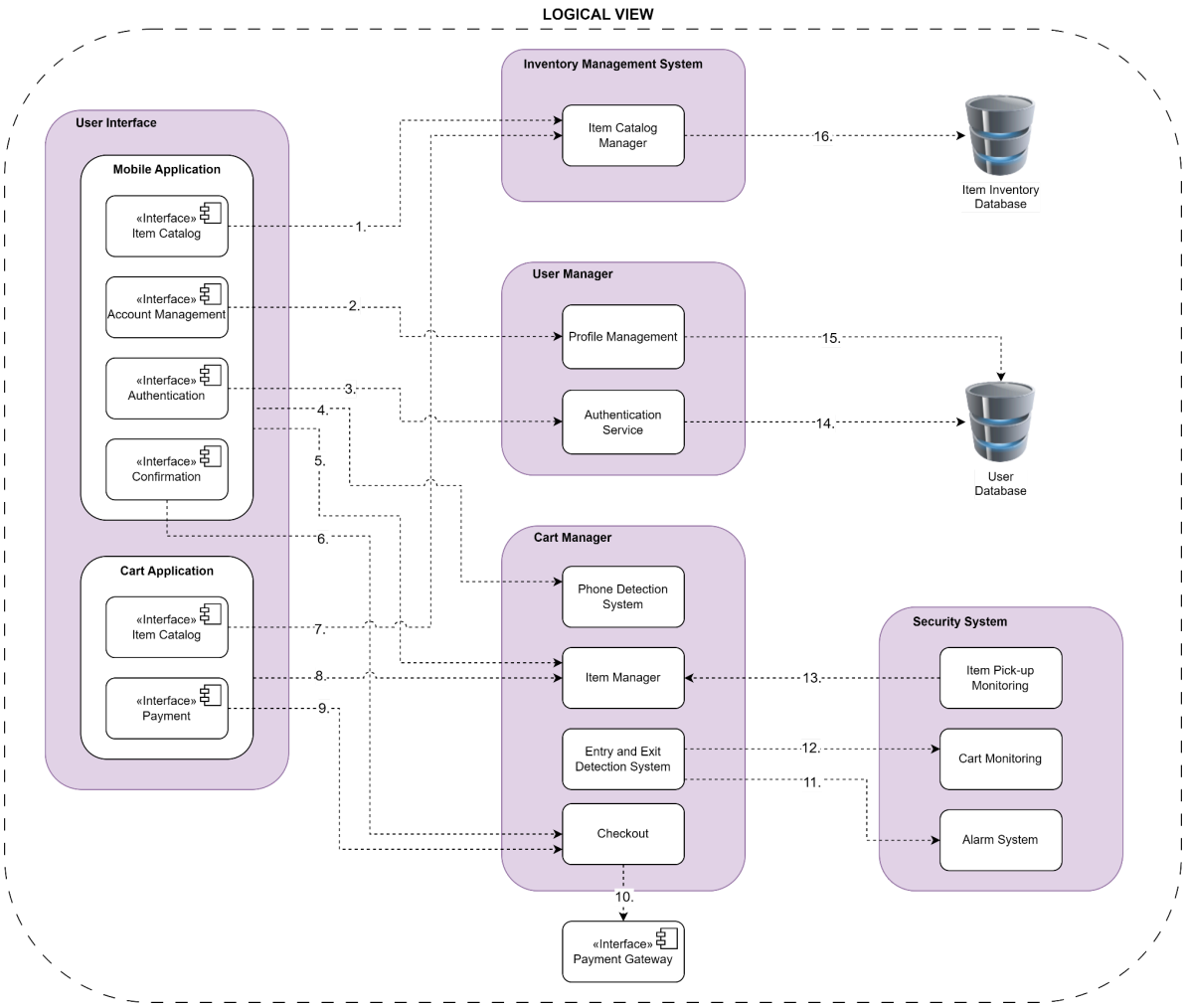

Figure 1. Logical View Diagram

Physical View

The physical view of the system outlines the tangible infrastructure components and their interactions. This view illustrates the distribution of system elements across hardware and network resources:

Customer

- **Mobile Phones**: The shopping cart system is accessible via mobile phones, catering to users who prefer the convenience and mobility of handheld devices and to pay via the app, without the need for a physical card. These users can connect to the application via mobile data or WiFi, which is then forwarded to the store's network to link to the closest Shopping card.
- **Shopping Cart**: Links to users using the application or to users who do not have the

application installed on their handheld, who can simply use the shopping cart, thus requiring the use of a physical credit/debit card. The shopping cart system is directly connected to the store's network via WiFi.

Internet

- The Internet serves as the primary medium through which the stores communicate with the global servers, to fetch user information and to offload the store's data backups.
- Clients use the Internet to enter the application and fetch their data, including available discounts and sales. Clients who use the store's local WiFi, are forwarded to the internet to do the above.
- Utilizing standard protocols and security measures, the Internet ensures reliable and secure data transmission between clients and servers.

Store Backend Infrastructure

- **Load Balancer**: Positioned at the forefront of the store's backend infrastructure, the load balancer efficiently distributes incoming traffic among multiple servers to optimize performance and maintain system reliability. It ensures that no single server becomes overwhelmed with requests, thereby preventing bottlenecks and ensuring scalability.
- **Server Cluster (N servers)**: The backend comprises a cluster of servers, with the number varying dynamically based on demand and load. Each server within the cluster hosts the application logic and processes user requests. The servers operate in parallel, collectively handling user interactions and system operations.
- **Store Database**: Each server in the cluster hosts a dedicated instance of the Store Database, which holds information related to the store's inventory. This includes information such as product name, stock, location in store, location in the warehouse, and pricing information. The database is designed for efficient retrieval and manipulation of store data, ensuring fast response times during user interactions in the store.

Global Backend Infrastructure

- **Load Balancer**: Positioned at the forefront of the backend infrastructure, the load

balancer efficiently distributes incoming traffic among multiple servers to optimize performance and maintain system reliability. It ensures that no single server becomes overwhelmed with requests, thereby preventing bottlenecks and ensuring scalability.

- **Server Cluster (N servers)**: The backend comprises a cluster of servers, with the number varying dynamically based on demand and load. Each server within the cluster hosts the application logic and processes requests. The servers operate in parallel, collectively handling interactions and system operations.
- **Stores Database**: Each server in the cluster hosts a dedicated instance of the Stores Database, which holds backup information related to each store's inventory.
- **Users Database**: Similarly, each server hosts a Users Database, responsible for managing user-related information such as user profiles, authentication credentials, purchase history, discounts, and sales. The User Database facilitates secure user authentication and authorization processes, ensuring that only authorized users can access functionalities such as purchase history and account management via the mobile application.

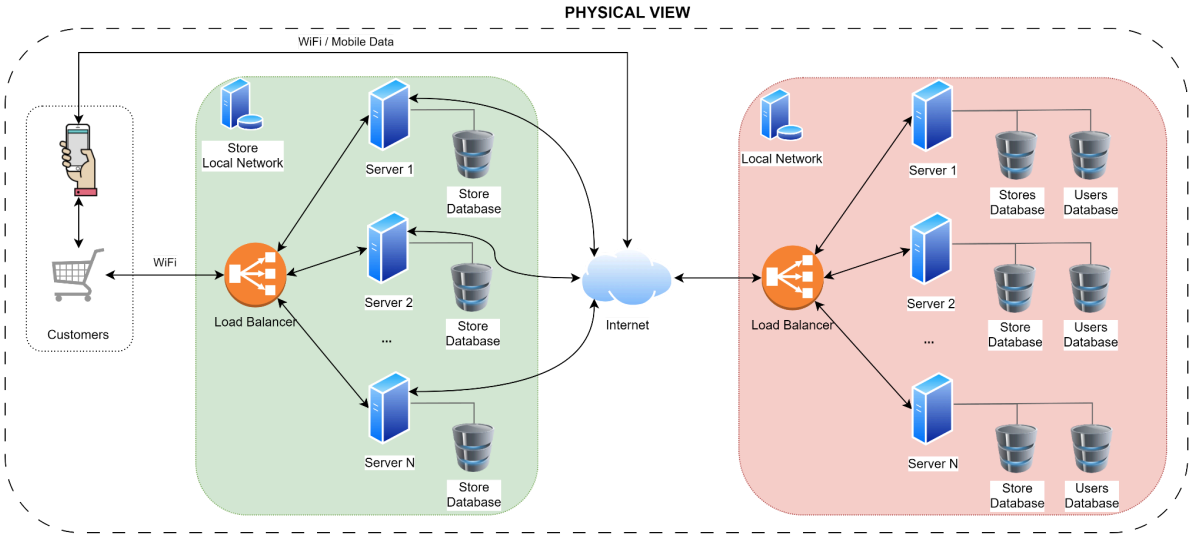

Figure 2. Physical View Diagram

Use-case View

In this section, we present some use cases and we will explore and describe the core elements of some of the scenarios. For each use case, we will start by identifying the title and description, to find the main goal of the use case and what it does. Next, we will identify the actors, i.e. the entities that interact with the use case, the preconditions, and the postconditions, meaning the conditions needed to be met before and after the use case, respectively. Lastly, we will identify possible paths for each use case, namely, a primary path with the most common route to success and an optional alternate path when a less common route is taken to success.

Some possible use cases for this system are the following:

1. User adds an item to the cart.
2. User walks out of the store (checkout).
3. User completes a purchase (with or without the application).
4. User leaves the store without paying.
5. User removes products from the shelves but never puts them in the cart.
6. User enters with a cart, but another user exits the store with that same cart. (Important for switching the phones identified by the cart).
7. User searches for a product using the application/cart device.
8. User can't pay for the products on exit. (How long should the system wait before triggering the alarm?)
9. User puts an item in the cart, but the system can't identify it.

For this report, we focus on these **9** use cases, but it's important to note that there are further use cases that can be important to understand how the system will operate fully.

The first **2** use cases contain a message sequence diagram, as requested on the assignment.

Use Case 1

Title: User adds an item to the cart.

Description: The user enters the store and adds an item to the cart.

Actors: User and System.

Preconditions: The user has the app.

Postconditions: Inventory is updated.

Paths:

- **Primary Path:**
    1) The user grabs a cart.
    2) The cart connects to the user's phone app.
    3) The user adds items.
    4) Cart updates total.
- **Alternate Path:**
    1) A user without an app grabs a cart.
    2) The cart connects to the user's phone app.
    3) The user adds/removes items.
    4) Cart updates total.

Priority and Frequency of Use: Top priority (main functionality) and very high frequency of use.

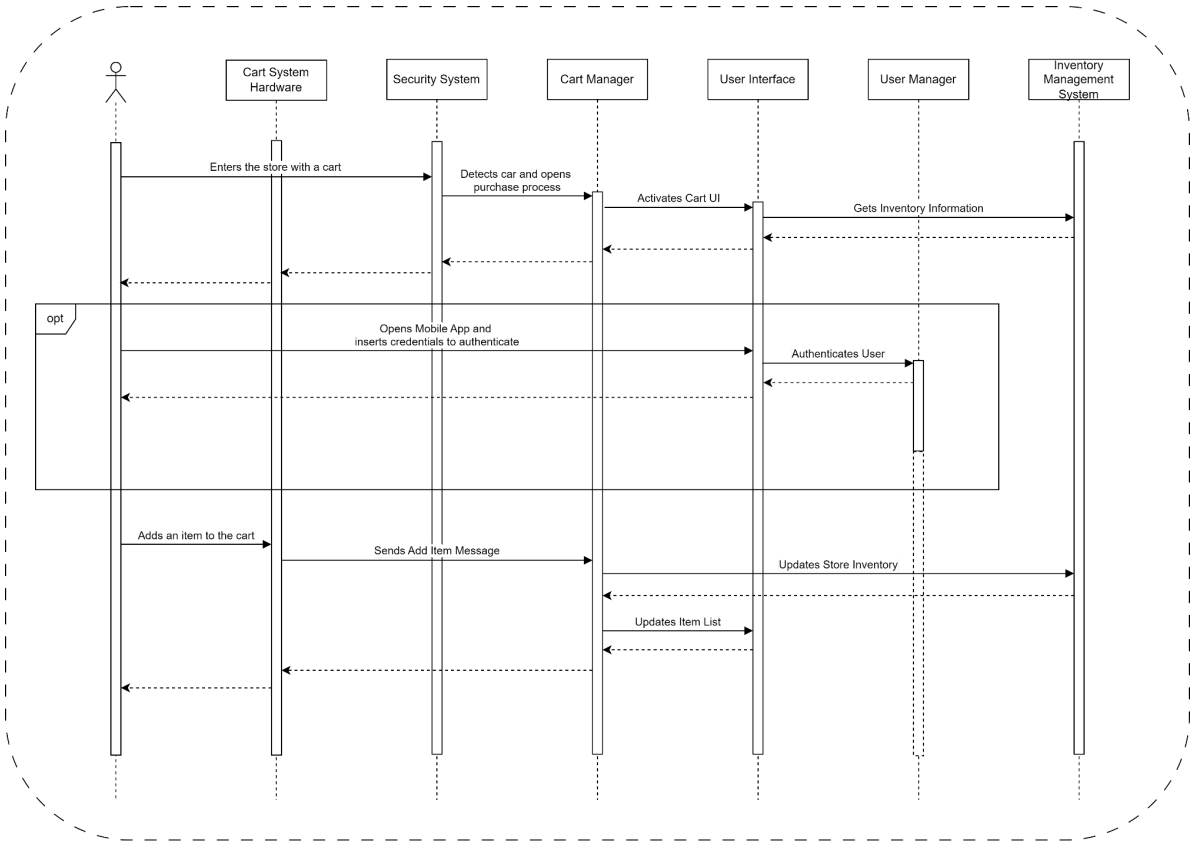

Figure 3. Use Case 1 - Message Sequence Diagram

Use Case 2

Title: User walks out of the store (checkout).

Description: The user enters, adds items to the cart, and proceeds to exit the store (checkout).

Actors: User and System.

Preconditions: The user has the app.

Postconditions: Payment is processed and inventory is updated.

Paths:

- **Primary Path:**
    1) The user grabs a cart.
    2) The user adds an item.
    3) Cart detects exit with items.
    4) User confirms payment method.
    5) Checkout is completed.

Priority and Frequency of Use: High priority and very high frequency of use.

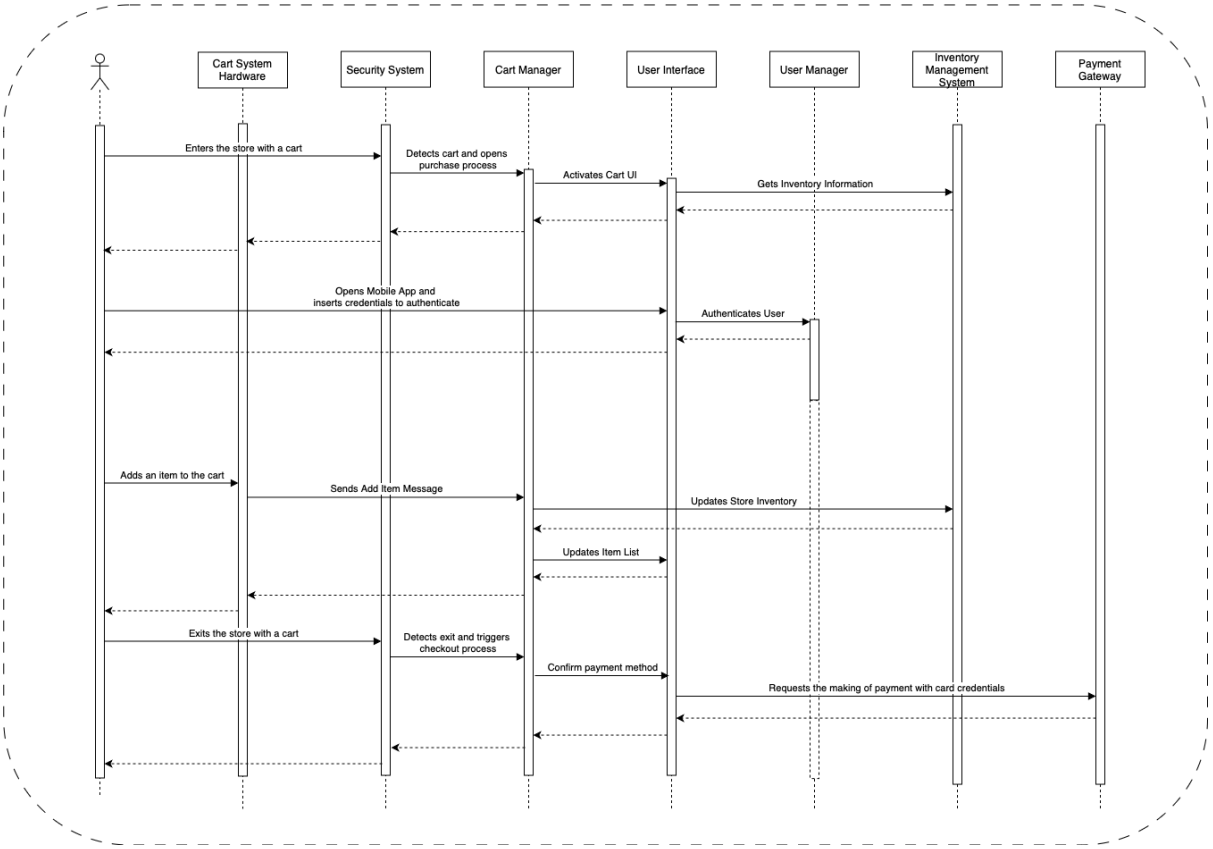

Figure 3. Use Case 2 - Message Sequence Diagram

Use Case 3

Title: User completes a purchase.

Description: The user enters the store and adds/removes items from the car throughout its visit to the store. At the end, the user exits and pays.

Actors: User and System.

Preconditions: Account created and application installed, if the user plans on paying through the app.

Postconditions: Purchase completed (payment done). The inventory is updated to no longer contain the items purchased.

Paths:

- **Primary Path:**
    1) The user grabs a cart
    2) The cart identifies the user's phone and connects to the app.
    3) The user adds products to the cart.
    4) The products are identified and the prices are added to the total.
    5) The user removes items from the cart.
    6) The products are identified and the prices are removed from the total.
    7) The user exits the store
    8) A prompt for payment appears on the application.
    9) The user pays using Venmo.
    10) The cart detects the payment and concludes the purchase
- **Alternate Path:**
    1) The user grabs a cart
    2) The user adds products to the cart.
    3) The products are identified and the prices are added to the total.
    4) The user removes items from the cart.
    5) The products are identified and the prices are removed from the total.
    6) The user exits the store
    7) A prompt for payment appears on the cart’s device.
    8) The user pays using a debit/credit card.
    9) The cart detects the payment and concludes the purchase

Priority and Frequency of Use: Top priority (main functionality) and very high frequency of use.

Use Case 4

Title: User leaves the store without paying.

Description: The user enters the store and adds/removes items from the car throughout its visit to the store. At the end, the user exits without paying, triggering the alarm.

Actors: User and System.

Preconditions: -

Postconditions: The alarm is triggered.

Paths:

- **Primary Path:**
    1) The user grabs a cart
    2) The user adds an item to the cart.
    3) The product is identified and the price is added to the total.
    4) The user exits the store.
    5) A prompt for payment appears on the cart's device.
    6) The user ignores the prompt and starts removing items from the cart.
    7) The security system detects the removal of items from the cart outside the store without payment and triggers the alarm.

Priority and Frequency of Use: High priority for security reasons with medium to low frequency of use.

Use Case 5

Title: User removes products from the shelves but never puts them in the cart.

Description: The user enters the store and holds onto items in his hand. At the end, the user exits without paying, triggering the alarm.

Actors: User and System.

Preconditions: -

Postconditions: The alarm is triggered.

Paths:

- **Primary Path:**
    1) The user picks up items from shelves.
    2) The user adds an item to the cart.
    3) Item Pick-up Monitoring Module detects items that were removed but not placed in any cart.
    4) The user exits the store with the cart empty.
    5) The security system detects if an item is outside the store without being

placed in a cart or paid, triggering the alarm.

Priority and Frequency of Use: High priority for security reasons with low frequency of use.

Use Case 6

Title: User enters with a cart, but another user exits the store with that same cart.

Description: The original User (User 1) enters the store. Before leaving the store, another User (User 2) takes hold of the cart. At the end, User 2 exits and pays.

Actors: User 1, User 2, and System.

Preconditions: The original User has the app connected to the cart.

Postconditions: Cart detects new user; recalculates bill if necessary.

Paths:

- **Primary Path:**
    1) User 1 enters the store and connects to a cart.
    2) User 2 takes over the cart.
    3) Phone Detection System detects new user's phone.
    4) User 1 bill is freed.
    5) User 2 continues shopping.
    6) User 2 leaves the store, paying for the items.
- **Alternate Path:**
    1) User 1 enters the store and connects to a cart.
    2) User 1 places discounted items into the cart.
    3) User 1 bill is calculated accordingly
    4) User 2 takes over the cart.
    5) Phone Detection System detects new user's phone.
    6) User 1 bill is freed.
    7) Bill is recalculated for User 2, which does not have the same discounts.
    8) User 2 continues shopping.
    9) User 2 leaves the store, paying for the items.

Priority and Frequency of Use: High priority for security reasons with low frequency of use.

Use Case 7

Title: User searches for a product using the application/cart device.

Description: The user checks on the application/cart where a product is. The system

returns the product location. The user finds the product.

Actors: User and System.

Preconditions: The user has the app or uses the cart device.

Postconditions: User finds product location.

Paths:

- **Primary Path:**
    1) The user inputs the product name in the app/cart device.
    2) The inventory Management System provides aisle numbers and shelves.
    3) The user locates the product in the store.
- **Alternate Path:**
    1) The user inputs the product name in the app/cart device.
    2) Inventory Management System returns that the item isn't in stock in the store.

Priority and Frequency of Use: High priority with a high frequency of use.

Use Case 8

Title: User can't pay for the products on exit.

Description: User attempts to leave the store with items. The system rejects the payment.

Actors: User and System.

Preconditions: User has items in the cart.

Postconditions: The system waits before triggering the alarm

Paths:

- **Primary Path:**
    1) The user attempts to pay at the exit.
    2) Payment Gateway declines the transaction.
    3) The system requires the User to try again.
    4) The system waits for a set time before alarm activation.

Priority and Frequency of Use: High priority with low frequency of use.

Use Case 9

Title: User puts an item in the cart, but the system can't identify it.

Description: The user places an item into the cart. The system can't identify the item. The system declines payment until the item is resolved.

Actors: User and System.

Preconditions: The user has items in the cart.

Postconditions: The item is either identified or the user is prompted

Paths:

- **Primary Path:**
    1) The user places an item into the cart.
    2) Item isn't recognized by the Item Manager.
    3) Item Manager warns the User to place the item again, insert item details, or seek assistance in-store.
    4) The user removes and adds the item again.
    5) Item is recognized by the Item Manager.
- **Alternate Path:**
    1) The user places an item into the cart.
    2) Item isn't recognized by the Item Manager.
    3) Item Manager warns the User to place the item again, insert item details, or seek assistance in-store.
    4) The user seeks assistance in-store
    5) Item is manually added to the cart.

Priority and Frequency of Use: High priority for security reasons with medium to low frequency of use.

Architecture Patterns

Layers

This pattern is prevalent in the presented architecture. SmartShoppingCart organizes the system into different layers, such as the presentation (user interfaces), application (user authentication and item detection), and data access layers. That promotes the solution's modularity and maintainability, as the concerns are separated among different layers.

Client-Server

Clients interact with the system via different kinds of user interfaces, sending requests to the server group for adding items to the cart, viewing its contents, and initiating the checkout procedure. This pattern involves consistency in the balanced distribution of data, which requires a lot of communication between clients and servers. This ensures scalability, fault

tolerance, availability, and efficiency.

Broker

This pattern eases the communication between components in a distributed system. Those can communicate without much detail, as the broker handles message routing and ensures reliability upon the delivery of requests.

Shared Repository

This pattern facilitates data sharing and consistency across different components of the solution since it involves centralized data storage that can be accessed by multiple parts. This technique is present in the items and users' databases, promoting data integrity.

Microservices Architecture

A microservices architecture forms the basis of the Smart Shopping Cart solution, which consists of several stand-alone services for handling carts, item management, user identification, and payment processing. With its database, each service functions independently, allowing for easy changes and scalability. This design supports fault isolation, maintainability, and scalability, making it a useful option for the Smart Shopping Cart system despite possible communication complexity and data consistency issues.

Quality Attributes - Requirements

- **Availability**: The system must be available to users with minimal downtime, especially during peak shopping hours. This quality attribute is affected by the scalability of the system, as it needs to maintain performance under an increased load of products and users.
- **Reliability**: This attribute is very important, as the system must be resilient to failures. Even considering software and hardware failures, consistent and reliable operations must be ensured, since the whole purpose of this system is to be practical, saving customers’ time.
- **Security**: The system must have reliable authentication methods, to prevent payment fraud. Whenever a user takes a shopping cart, they must scan the QR code in it to connect to the app, which ensures that payments are only relative to the user’s cart, and no other.

- **Usability**: Although the user interface will be minimalist, it must be user-friendly, since it is expected that users of all ages will use the system. That will minimize the time required for users to complete tasks and boost their preference over traditional methods.
- **Maintainability**: This attribute is also connected to modularity since the system's development must follow a modular approach, as it is crucial to ease maintenance and future development. It also boosts availability by minimizing the time required to solve possible problems.
- **Interoperability**: This is related to the integration with external systems. The solution must be designed in a way that enables the seamless integration of hardware, in this case, payment systems and shopping carts. An optimized integration enhances availability and general performance.
- **Testability**: Testability is essential to enable comprehensive testing of the functionalities. The validation of system behaviors is obtained with mocking and stubbing, which allows to isolate dependencies. That ensures the good functioning of the solution.

These quality attribute requirements are foundational to ensuring that the Smart Shopping Cart system delivers a reliable, high-performance shopping solution that meets the evolving needs of today's customers and supermarket owners.

Conclusion

The Automated Smart Shopping Cart architecture offers a complete solution for managing automatic, hassle-free physical shopping experiences. Through a modular architecture, the system efficiently and effectively handles user interactions, insertion and removal of items from the shopping cart, and transactions. The multiple views showcase several important features of the system, including key interactions between components. Additionally, detailed use-case scenarios address various user interactions and potential conflicts. The architecture emphasizes availability, reliability, and security, catering to specific requirements for such a system. Overall, the architecture has a robust design for an efficient and fast shopping experience.