Advanced Programming

Workshop No 2 – Object-Oriented Programming

Mariam Betin Escobar

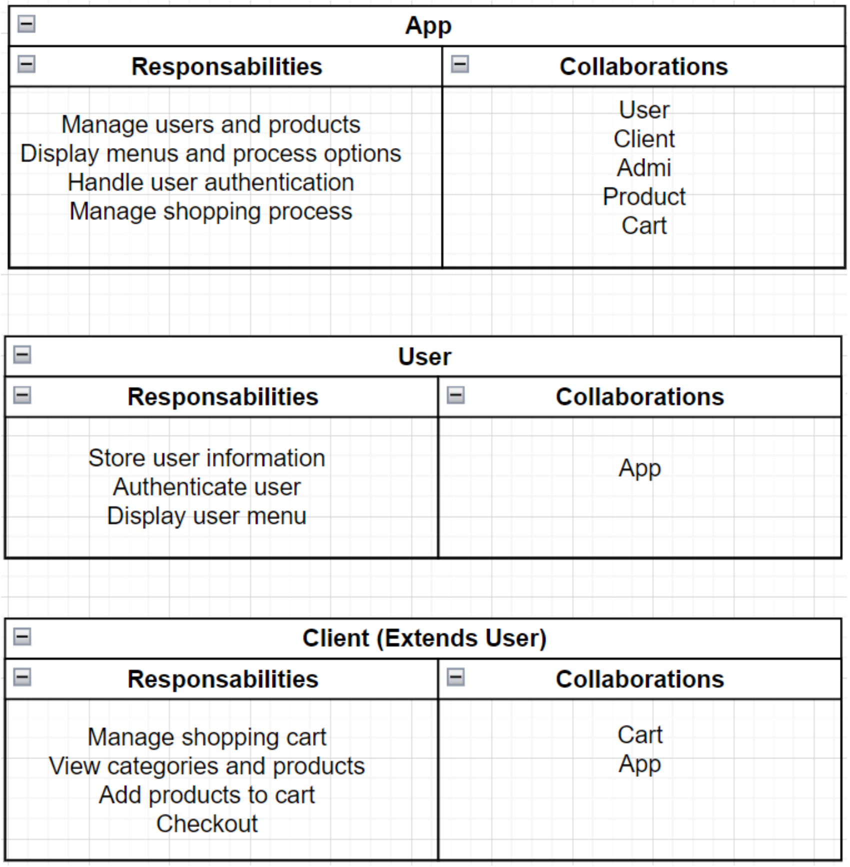
Universidad Distrital Francisco Jose de Caldas

05-10-2024 III

An update and refinement of the system software made to the electronic device store is presented. Where the project has a detailed analysis and significant improvements have been implemented.

1. **Refinement of user stories:**
   We surveyed more people and from different specializations, thus getting several points of view, and found the following:
   - As an administrator, I want to be able to add new products so I can expand the catalog.
   - As an administrator, I want to be able to remove products from the catalog when we can no longer sell a device.
   - As a user, I want to be able to see reviews from other buyers to make informed decisions.
   - As an administrator, I want to be able to update the stock of the products in case any of them is on sale and to be able to have additional in the sale.
   - As a buyer, I want to compare features of similar devices side by side to rank them according to my preferences.
   - As a customer, I want to see detailed technical specifications of the devices to be able to compare according to what I want.
   - As an administrator, I want to be able to see all the products in the catalog to manage the inventory as fast as possible.

2. **CRC carts:**
   These cards provide an efficient way to identify, organize and represent the classes, along with their responsibilities and the interactions they have with each other, where:



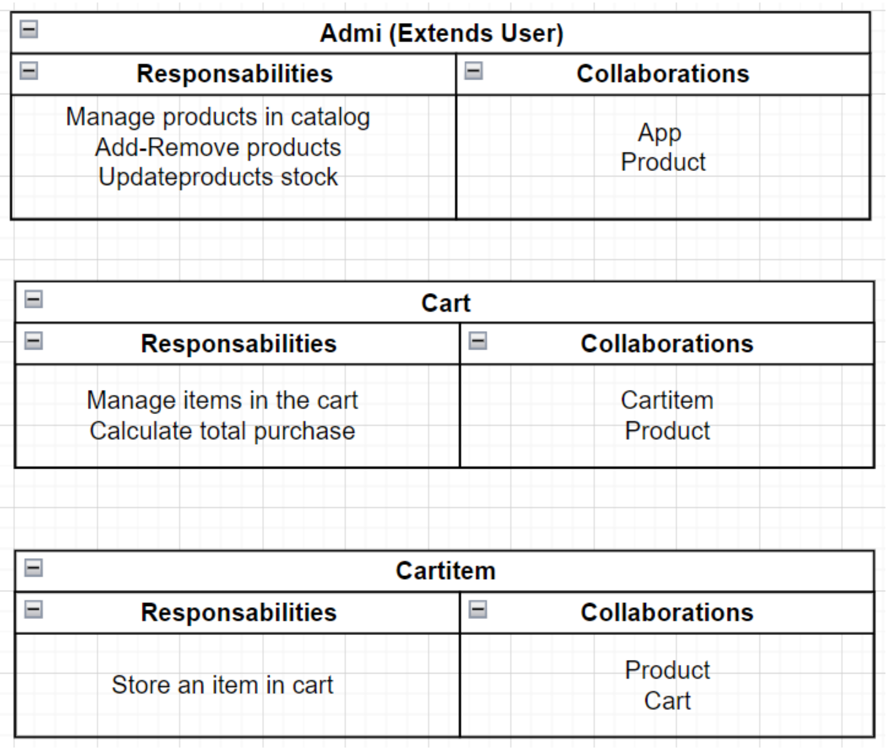
App	
Responsabilities	**Collaborations**
Manage users and products Display menus and process options Handle user authentication Manage shopping process	User Client Admi Product Cart

User	
Responsabilities	**Collaborations**
Store user information Authenticate user Display user menu	App

Client (Extends User)	
Responsabilities	**Collaborations**
Manage shopping cart View categories and products Add products to cart Checkout	Cart App

The App is the logic of the system, allowing interaction between users and available products.

The User has the common information of all system users, providing a framework for authentication and management of essential user data. From this class, subclasses are derived that allow specialized management of the functionalities specific to each type of user, such as: Client, which represents the users that interact with the system as customers, allowing them to add products to their shopping cart, visualize categories and available products, and perform transactions



Admi (Extends User)	
Responsabilities	**Collaborations**
Manage products in catalog Add-Remove products Updateproducts stock	App Product

Cart	
Responsabilities	**Collaborations**
Manage items in the cart Calculate total purchase	Cartitem Product

Cartitem	
Responsabilities	**Collaborations**
Store an item in cart	Product Cart

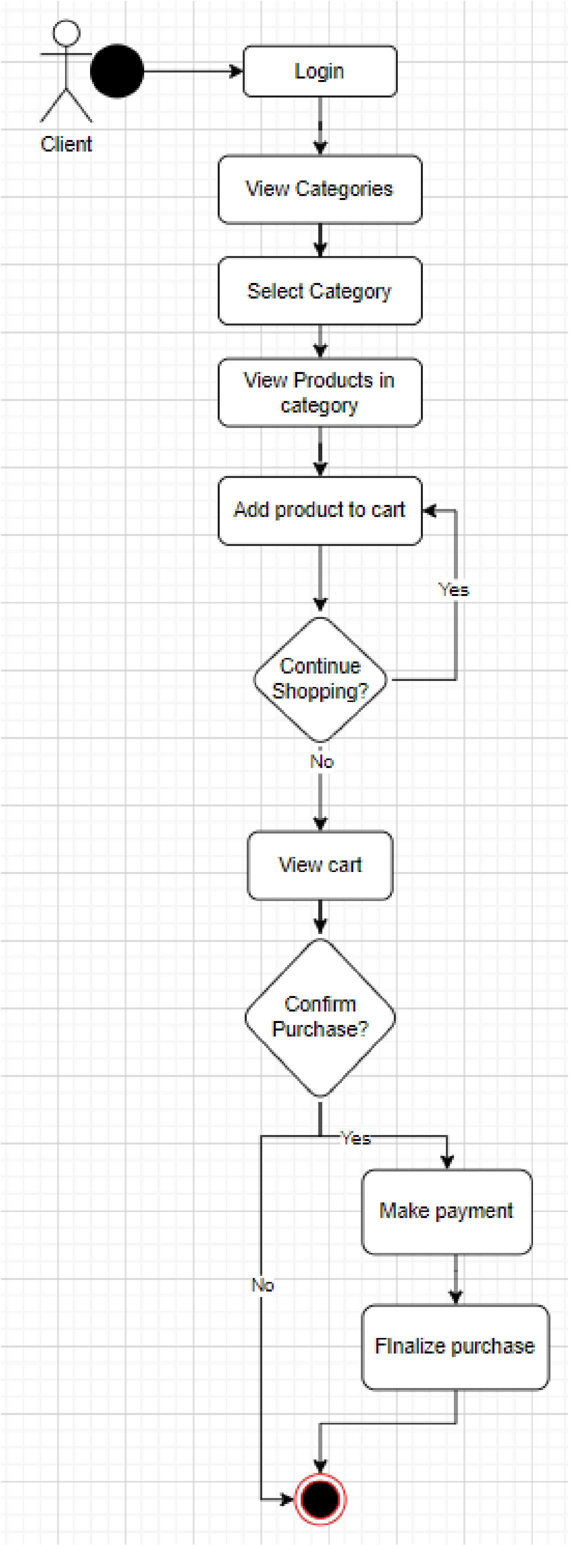
While the Admi class allows administrators to manage products, update inventories and add or remove products from the system.

The product system is managed by the Product class, which organizes product-related information and ensures that products can be efficiently managed in the inventory.

And the Cart and Cartitem classes organize and manage customer shopping carts. The Cart manages multiple items in a purchase, represented by Cartitem, with specific information about each product added to the cart.

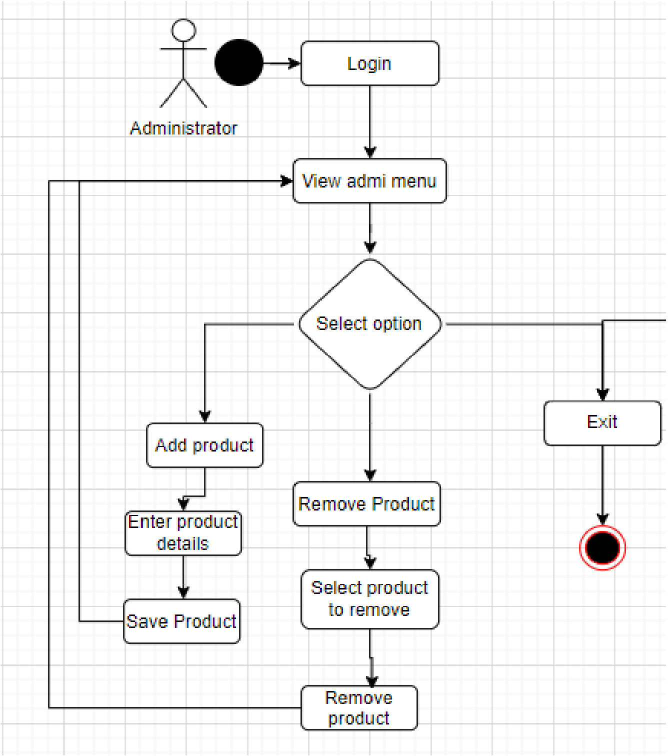
3. Activity Diagrams:

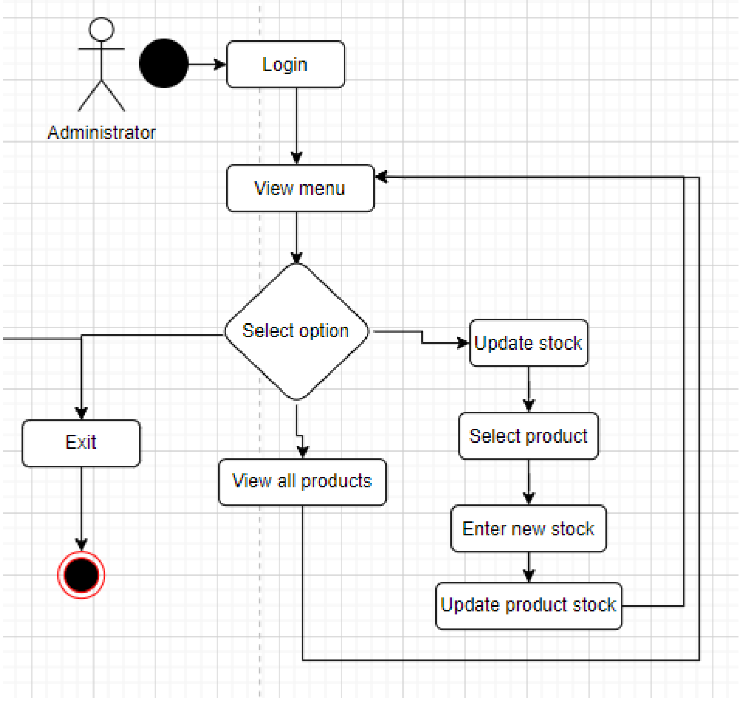
This diagram shows the flow of activities that a customer can perform, from login to completion of a purchase. Whit steps such as viewing categories, selecting products, adding to cart and checking out.



This diagram illustrates the activities that an administrator can perform, such as adding new products, deleting products, updating inventory and viewing all products in the catalog.

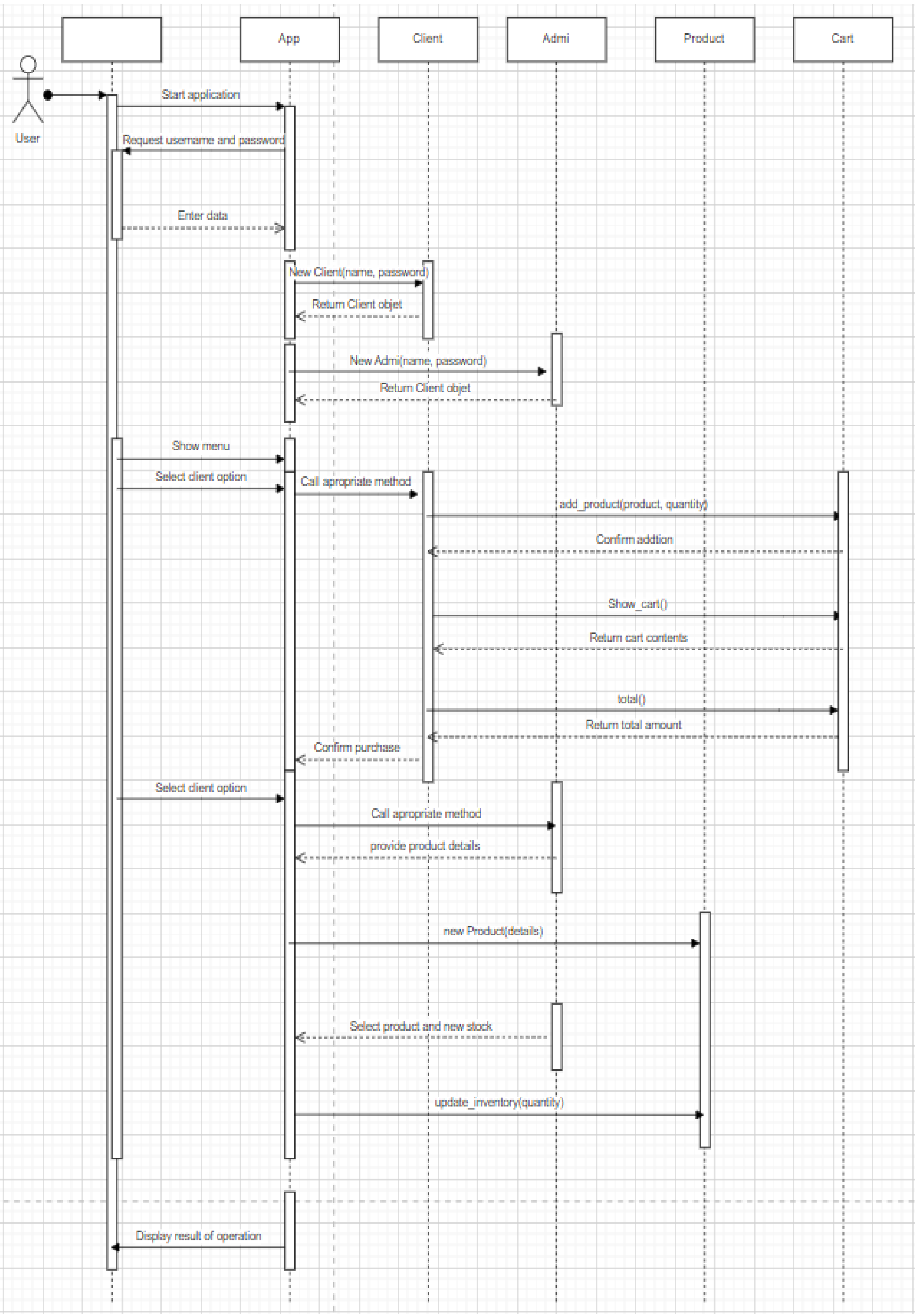
It provides a clearer view of the main processes of the system, making it easier to understand how different types of users can interact differently with the application.





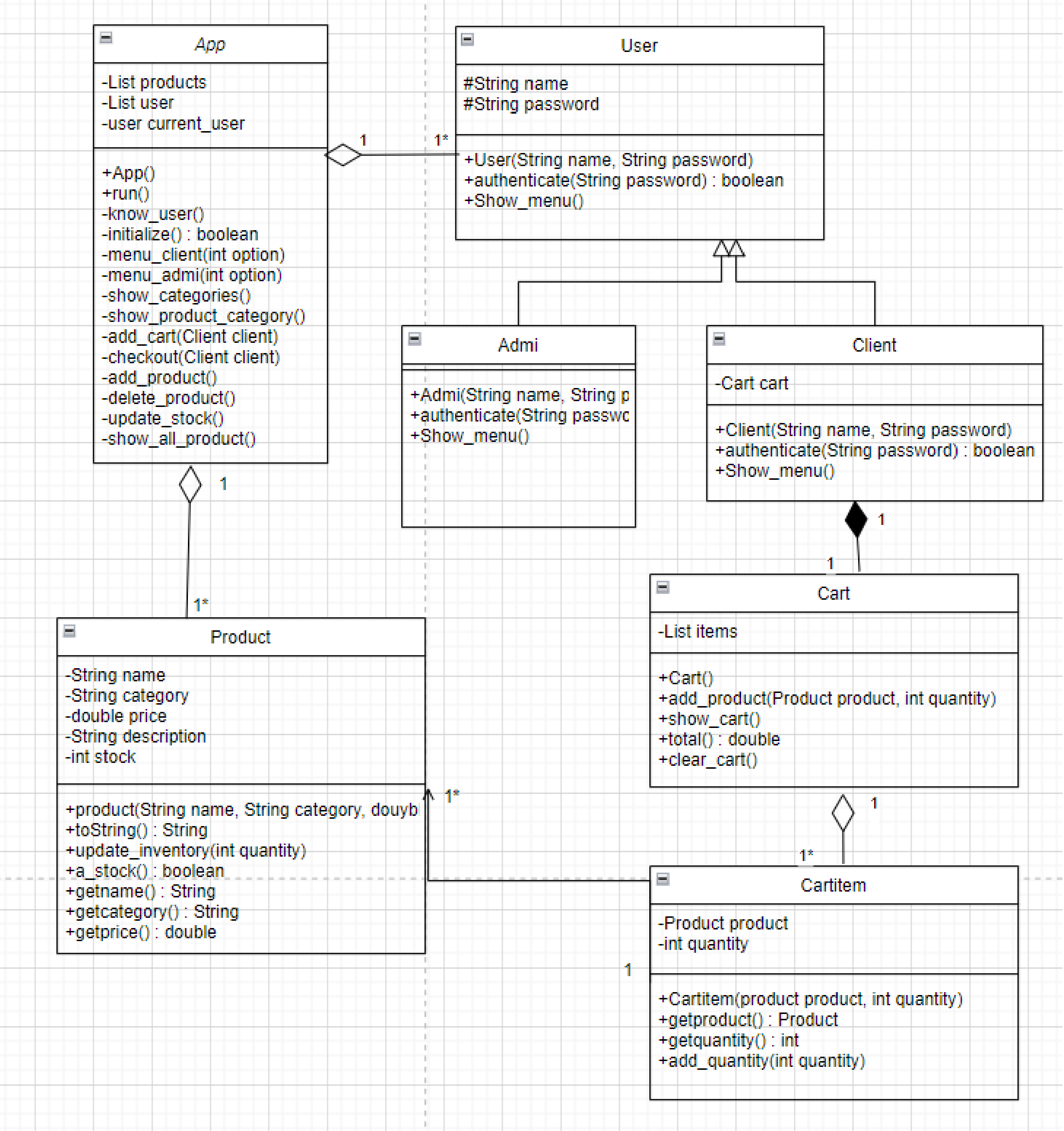
4. Sequence Diagrams:

This diagram shows how the different classes (App, User, Client, Admi, Product, Cart) communicate during processes such as user authentication, product selection and purchasing.



5. Class Diagrams:

This diagram includes the main classes such as App, User, Client, Admi, Product, Cart and CartItem, and shows how they relate to each other, allowing you to see the overall structure of the system.



6. **Changes and Enhancements:**

   The changes that were presented generated improvements to make the code more organized, maintainable and functional, in order to significantly improve the user experience compared to the initial version.

   - Object Oriented Architecture and Design:
     A more robust structure with specific classes was implemented. This improved code organization and separation of responsibilities.
   - Authentication System and User Roles:
     An authentication system with differentiated roles for clients and administrators was introduced. Each user type has its own menu and specific functionalities.
   - Improved Product Management:
     A more complete Product class was created with functionalities to add, delete, update and display products by category without knowing the products as they may vary.
   - Functional Shopping Cart:
     A more robust cart system was implemented with the Cart and Cartitem classes. It includes functionalities to add products, improving functionalities in the cart content, calculate total.
   - In-Memory Data Management:
     Although a database is not implemented, in-memory data management was improved. Products and users are stored in lists within the App class, maintaining data consistency during execution.

The system meets the basic required functionalities. But, the need to implement improvements in areas such as data persistence and system scalability has been identified. These improvements will allow the system to grow and adapt to a greater number of users and products.