

## Advanced Programming Workshop #1

Sebastian Avendaño Rodriguez

### Version 1.0

The application consists of simple system for an Electronic Device business, the main requirements were:

- a. Showing all the categories based on the user stories
- b. A working shopping cart
- c. Checkout paper
- d. One side-requirement that I get from one user story

The process was good and smooth, I need to improve some classes and relationships to get less code, also, need to see if I can get more relations in terms of usage SOLID principles. I need to improve packaging and modules with Poetry and the actual environment.

### Business Model:

In this platform, the user can get the catalog of electronic devices (got a lot of categories). They can interact with the reviews and other upcoming features

### Business rules:

Need to work on this, they only got the review system, but none can interact directly with them, just upload them and see them.

(More rules in the next versions)

### USER STORIES

You can see them in the docs/userStories.md. But, on the other hand, these are the user stories.

1. I, as a customer, want to see **dataphones** in the catalog so that I can easily purchase a mobile payment device.
2. I, as a customer, want to browse **washing machines** in the catalog and see user reviews so that I can find a reliable appliance for laundry with trustworthy feedback.
3. I, as a customer, want to see a variety of **phones** in the catalog so that I can choose the right smartphone for my needs.
4. I, as a customer, want to see **headphones** in the catalog with a comparison tool so that I can compare sound quality and price.

5. I, as a customer, want to view **microwaves** in the catalog so that I can buy a fast and convenient cooking appliance.
6. I, as a customer, want to get **clothes irons** and get opinions based on popular features so that I can easily select the best iron for my clothes.
7. I, as a customer, want to browse **PCs** in the catalog.
8. I, as a customer, want to see **TVs** in the catalog.
9. I, as a customer, want to see various **blenders** in the catalog with a rating system so that I can decide based on customer experiences.
10. I, as a customer, want to browse **air fryers** in the catalog so that I can choose a healthier alternative for cooking fried food.
11. I, as a customer, want to see **pressure cookers** in the catalog with recipe suggestions so that I can buy one knowing I'll have ideas for meals.
12. I, as a customer, want to browse **Xbox Series X** consoles in the catalog so that I can purchase a next-gen gaming system.
13. I, as a customer, want to see **PlayStation 5** (PS5) consoles in the catalog so that I can decide if it's the right gaming system for me.
14. I, as a customer, want to view **laptops** in the catalog with specs comparison, so that I can easily compare features like battery life and storage.
15. I, as a customer, want to see **smartwatches** in the catalog so that I can purchase a wearable device to track my fitness and notifications.

## Entities

- User (the actual customer)
- Catalog (Categories)
- Shopping cart
- Reviews
- Checkout receipt

## OOP PRINCIPLES ANALYSIS

Well, this is kinda hard to do, I tried to implement SOLID principles in the shopping cart class using an abstract class.

Polymorphism can be added in the ShowCategories classes, using the method open categories, probably this will be added.

Inheritance was used by creating an abstract class for the implementation of SOLID principles (I tried to)

Encapsulation it's hard to do in Python due to the lack of "private", "protected" or "public" management in the classes, so this part can be skipped

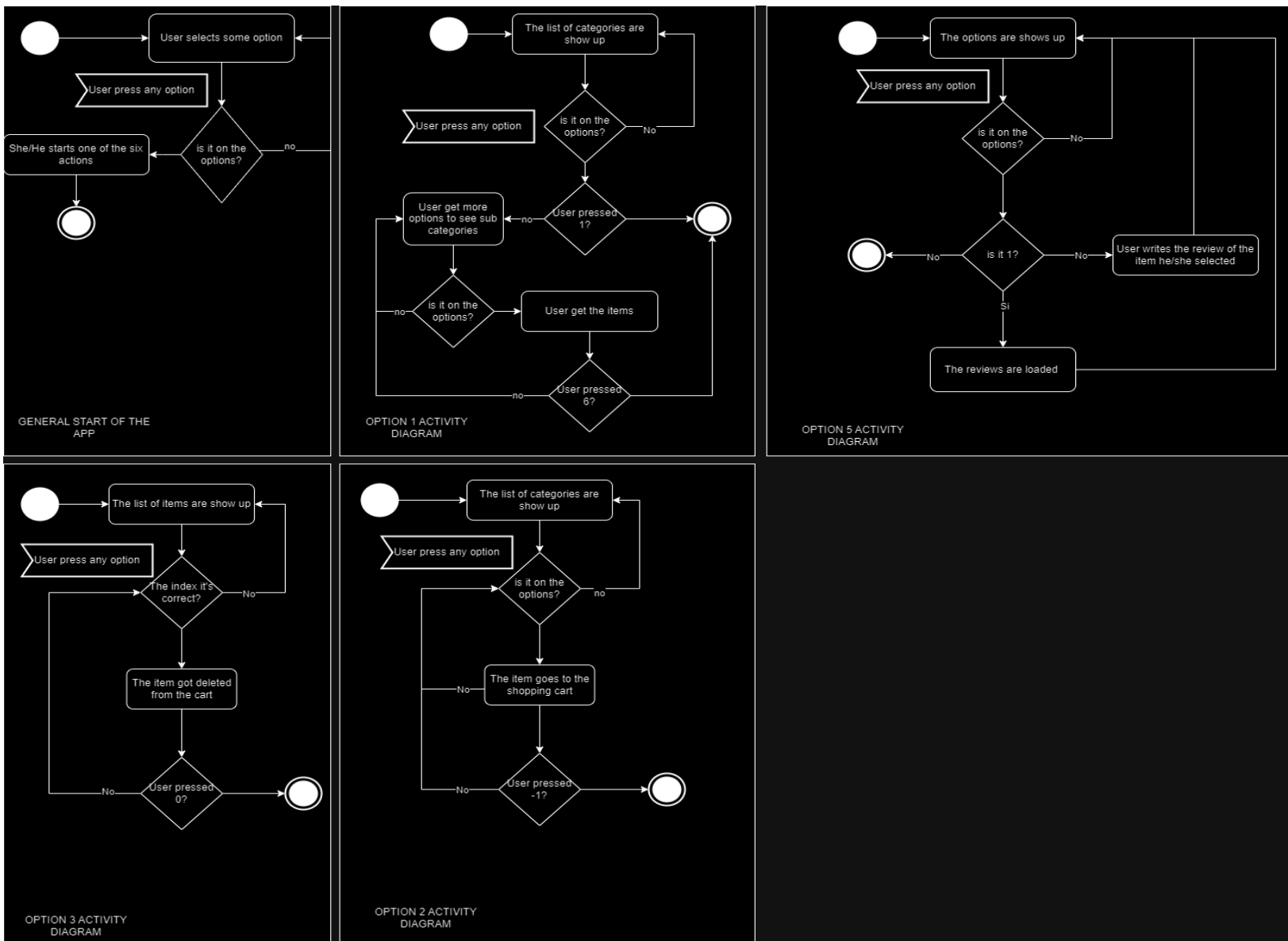
Abstraction: I think I lacked a lot in this part, because in the main.py there is a lot of code that somehow can be in a class or classes, I need to look up on this. Maybe, this happened because of my indiscipline doing the code and the preparation, but can be corrected (I hope)

## CRC CARDS

ShowCategories	
Responsability Provides all the catalog information	<u>Collaborators</u>
Reviews	
Responsability  Provide: - Adding items reviews into a simulated storage - Writing and reading the information - Giving users reviews to other users	<u>Collaborators</u>  <u>ShowCategories</u>
ShoppCart	
Responsability  Provide: - Adding items into the shopping cart - Adding the total money to pay - Removing items from the shopping cart	<u>Collaborators</u>  <u>ShowCategories</u>

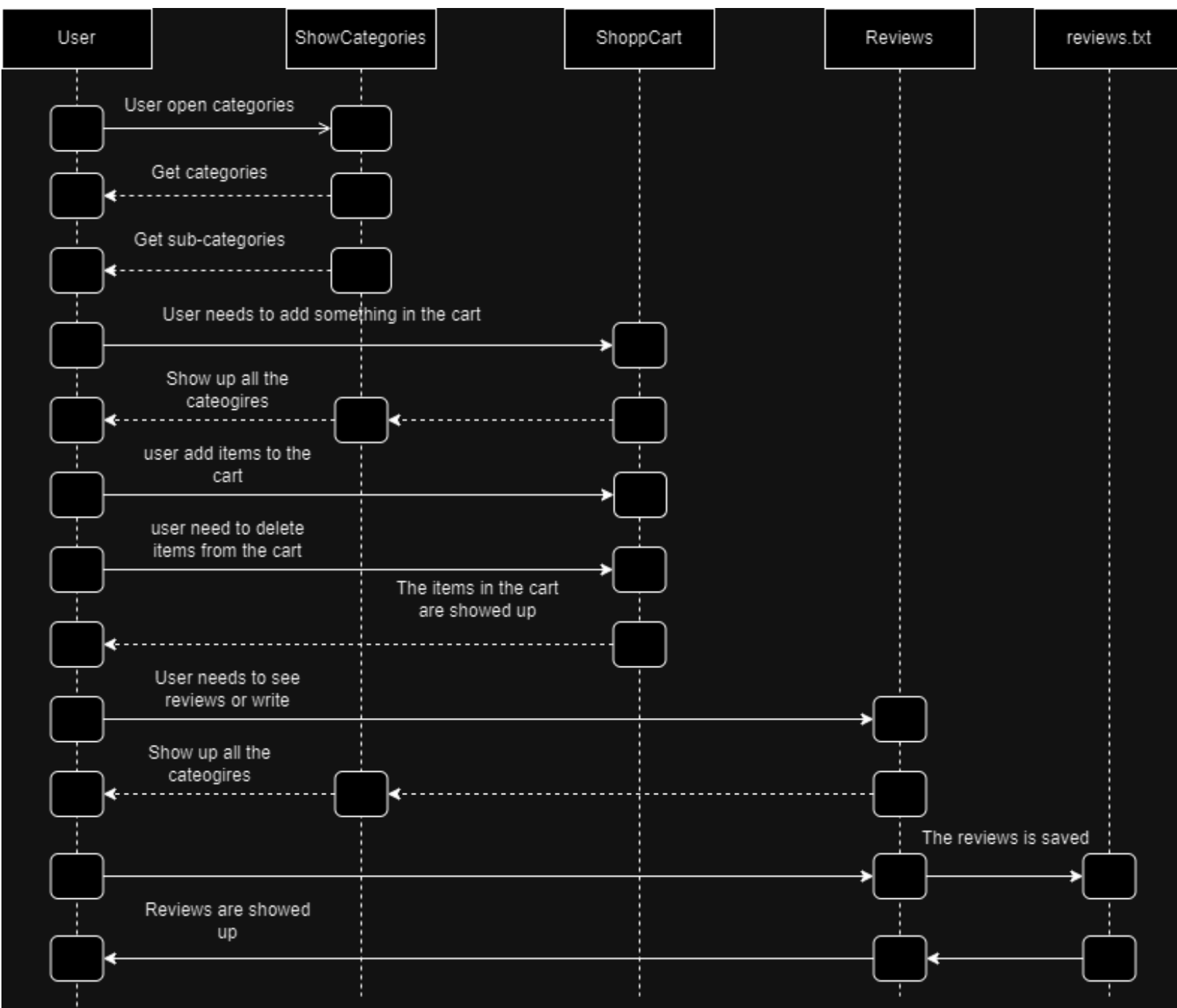
**Note:** I can add more CRC cards in the future, due to this is the Version 1.0 of the system

## ACTIVITY DIAGRAMS



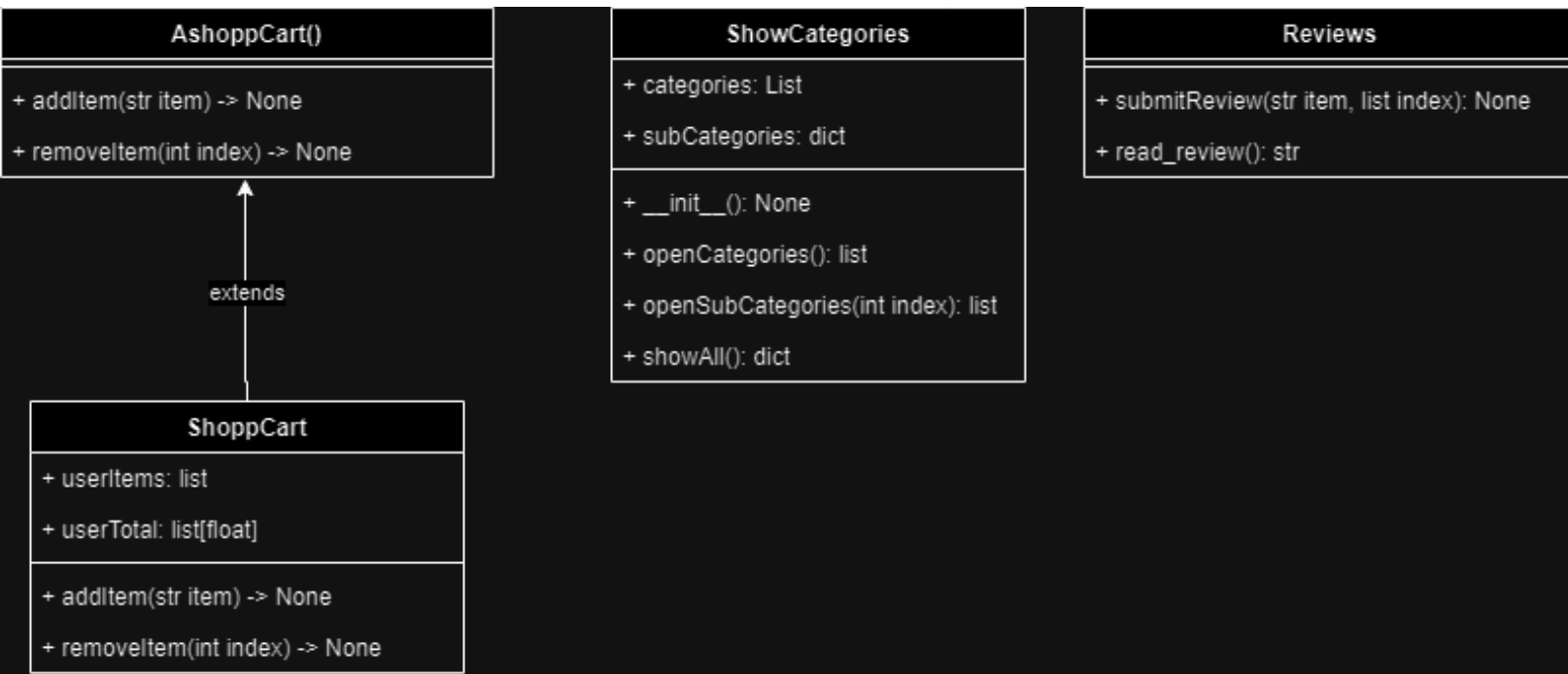
**Note:** The check-out generator has no activity diagram because there are simple steps to finish that thing

## SEQUENCE DIAGRAM



**Note:** The whole diagram can change, if the steps and activities are reduced or simplified

## CLASS DIAGRAMS



**Note:** The whole diagram can change due to the poor relations and the implementation of more classes and possible relationships