Software Modeling I

Season 2024-III

Report

Workshop No. 1 — Object-Oriented Programming

Anderson David Arenas Gutierez – 20231020030

User Stories

User stories help capture the needs and expectations of end users in simple, understandable terms. Below are some of the user stories implemented in the arcade machine purchasing program:

1. **As a retro gamer**, I want to include *Pac-Man* in my arcade machine, so that I can relive the classic gaming experience I enjoyed as a kid.
2. **As a parent**, I want to choose games that are family-friendly, so that my kids can enjoy the arcade machine safely.
3. **As a collector**, I want to see the total price of the arcade machine before making a final decision, so that I can budget properly for my gaming setup.
4. **As a competitive gamer**, I want to ensure *Street Fighter* is in my arcade machine, so that I can practice and improve my skills for tournaments.
5. **As a tech enthusiast**, I want to select the arcade machine made of a good material, so that I can have the most durable and lightweight option available.
6. **As a buyer**, I want the ability to review all my selected games before completing the purchase, so that I can make sure I haven't missed any titles I want to play.
7. **As a casual gamer**, I want to add only a few games like *Tetris* and *Frogger*, so that I can enjoy short, relaxing game sessions without being overwhelmed by too many options.
8. **As a student on a budget**, I want to compare the price differences between materials, so that I can make the most affordable choice for my arcade machine.
9. **As a vintage gaming fan**, I want to see *Space Invaders* as a game option, so that I can recreate the nostalgic feel of an old-school arcade in my home.
10. **As a gift giver**, I want to create a custom arcade machine with games my friend loves, like *Donkey Kong* and *Galaga*, so that I can give them a personalized gaming experience.

11. **As a gamer**, I want to be able to switch my material choice if I accidentally choose the wrong one, so that I don't end up buying a machine made from a material I don't want.
12. **As a new buyer**, I want the program to notify me if I accidentally select the same game twice, so that I don't spend extra money on duplicate games.
13. **As a busy person**, I want the option to finish selecting games quickly and proceed to check out, so that I don't spend too much time making decisions.
14. **As a detailed-oriented buyer**, I want the program to display a clear summary of my machine, including the material, games, and total price, before finalizing the purchase, so that I can be sure everything is correct before committing.
15. **As a curious customer**, I want to see the prices for each game in the catalog, so that I can decide whether adding an extra game like *Centipede* is worth the cost.

Object-Oriented Principles Analysis

The system is designed following some of key object-oriented programming (OOP) principles. Here's how each principle has been applied:

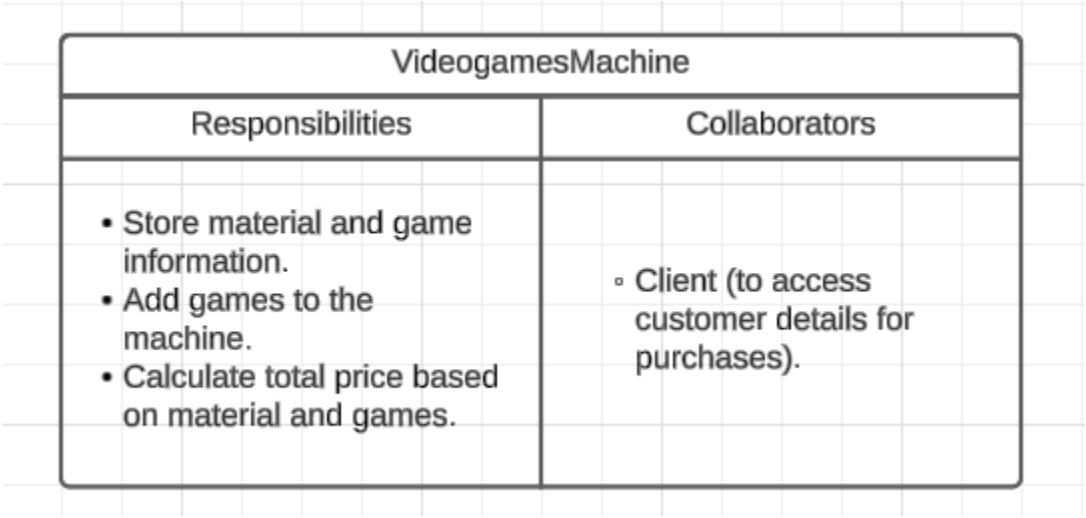
- **Encapsulation**: The internal details of the classes, such as attributes and methods, are hidden from the external world. For instance, the VideogamesMachine and Client classes handle specific attributes like games, material, name, and address, which are accessed and modified through public methods.
- **Abstraction**: The system abstracts the arcade machine and the client as objects with relevant attributes and behaviors. Users don't need to know the internal workings, they simply interact with the CLI to select a material, games, and see the total price.

However, here are the options on how to add the other principles:

- **Inheritance**: Although the current system does not explicitly use inheritance, future extensions could use inheritance to define different types of arcade machines that inherit common functionality from a base Machine class.
- **Polymorphism**: While polymorphism is not heavily used, methods like CalculatePrice could be overridden in subclasses if more types of machines were introduced, allowing different pricing calculations depending on the machine type.

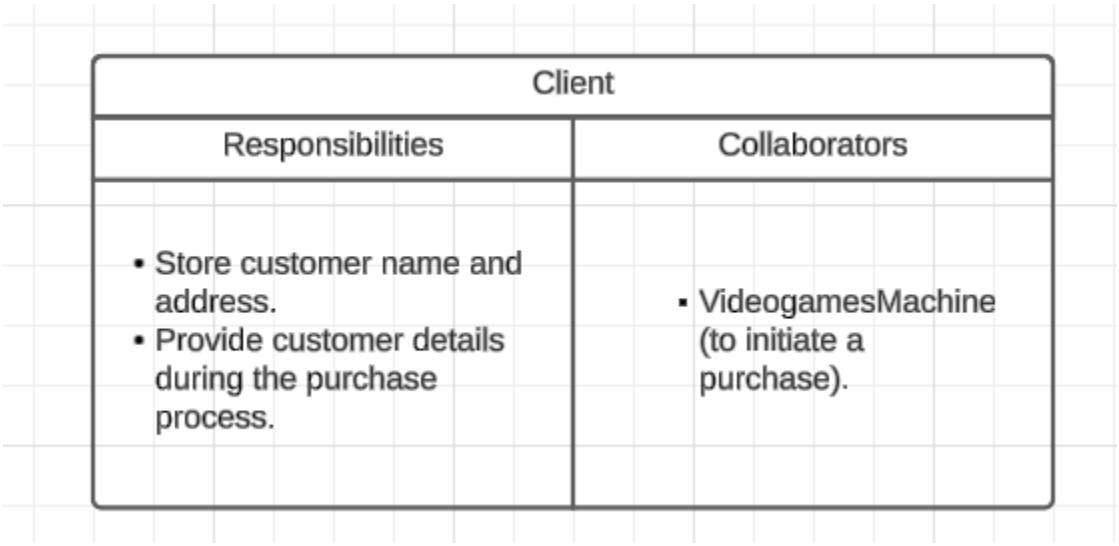
CRC Cards

- VideogamesMachine



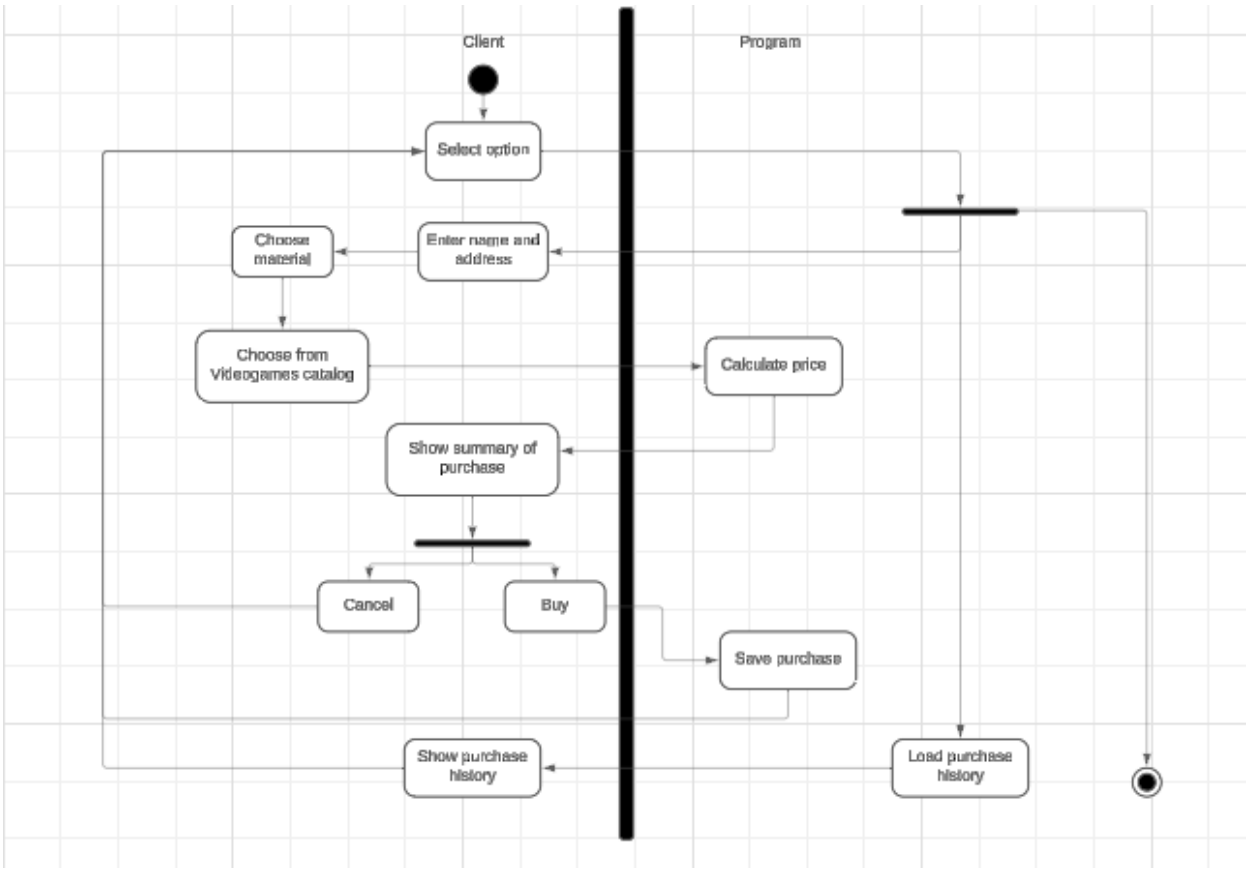
VideogamesMachine	
Responsibilities	Collaborators
• Store material and game information. • Add games to the machine. • Calculate total price based on material and games.	▫ Client (to access customer details for purchases).

- Client

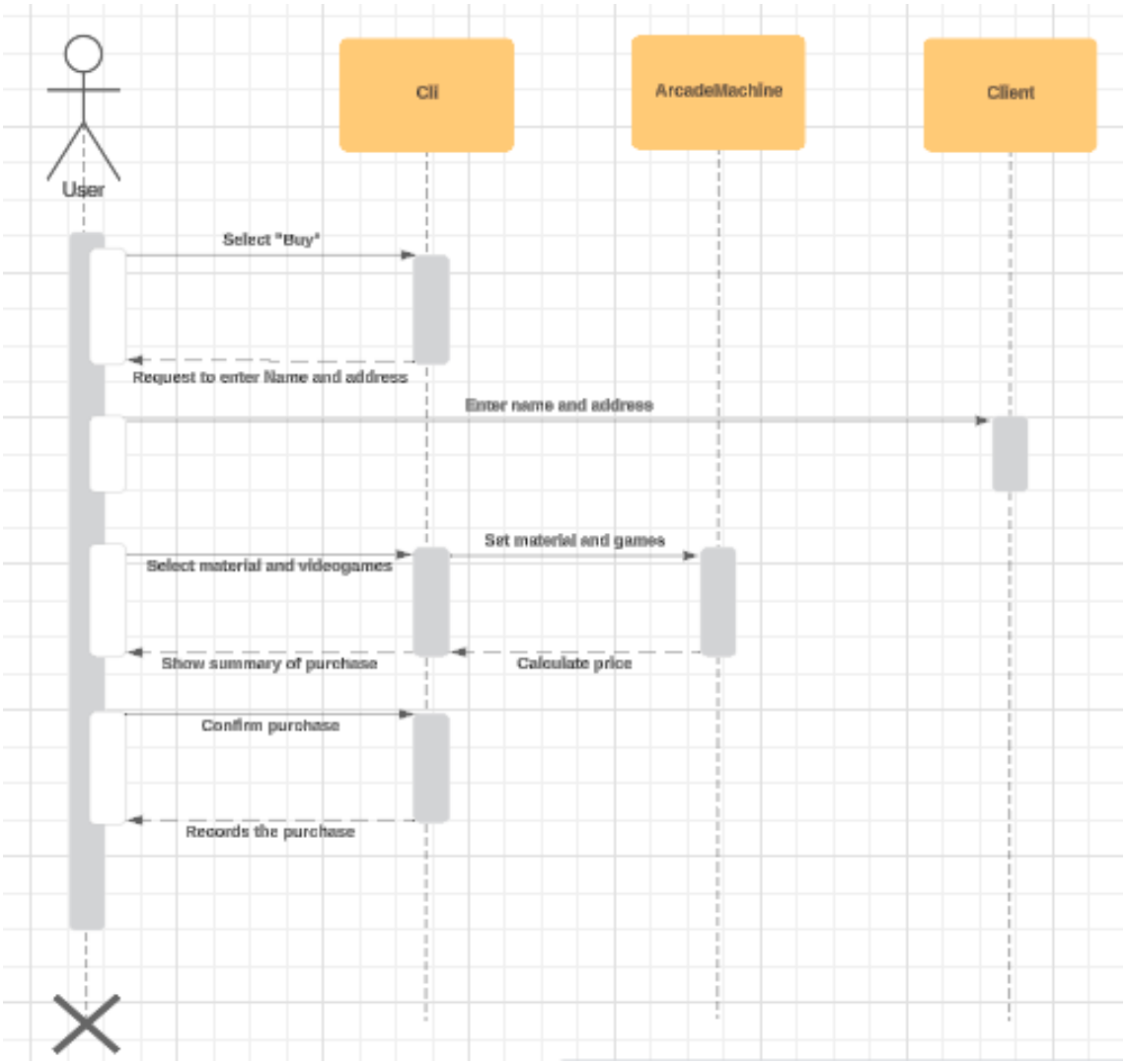


Client	
Responsibilities	Collaborators
• Store customer name and address. • Provide customer details during the purchase process.	▪ VideogamesMachine (to initiate a purchase).

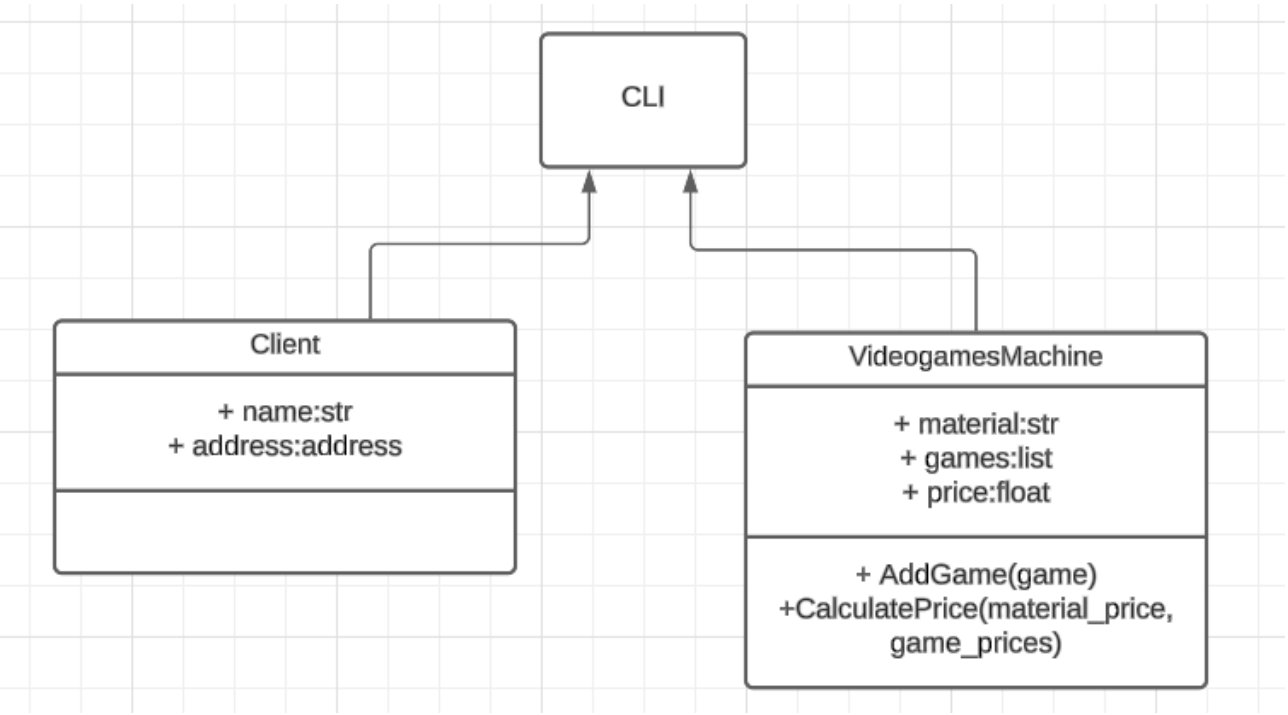
Activity diagram



Sequence diagram



Class diagram


CLI
Client
+ name:str
+ address:address
VideogamesMachine
+ material:str
+ games:list
+ price:float
+ AddGame(game)
+CalculatePrice(material_price,
game_prices)