

Report on the Development of an Arcade Video Game Machine Application

Santiago Alejandro Guada Bohorquez

Software Modeling I

Carlos Andrés Sierra

Universidad Distrital Francisco José de Caldas

September 14 , 2024



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

1. Introduction

This report outlines the development process of an application for managing arcade video game machines, focusing on the features and functionalities desired by users. Surveys were conducted with students to identify their preferences, and user stories have been formulated to guide the application's development.

2. Student Survey

Surveys were conducted with 15 students passionate about video games, yielding the following opinions:

- Customization of machine design.
- List of retro games.
- Connectivity options (Bluetooth).
- Inclusion of game reviews.
- Choice of control types.
- Touch screen interface.
- Information on battery life.
- Reward system for playing.
- Sound customization options.
- Flexible payment options.
- Section for recommended machines.
- Demo videos for each game.
- Multiplayer mode.
- Warranty and after-sales service options.
- Accessible customer support system.

3. User Stories

Based on the responses collected, the following user stories were created:

1. As a gamer, I want to customize the design of the machine, so it fits my style.
2. As a gamer, I want to see a list of retro games, so I can choose my favorites.
3. As a gamer, I want the machine to have connectivity options, so I can play with my friends.
4. As a gamer, I want to read game reviews, so I can find out which games are the most popular.
5. As a gamer, I want to choose the type of controls, so I can have a better gaming experience.
6. As a gamer, I want the machine to have a touch screen, for easier navigation.
7. As a gamer, I want to know the battery life, so I can use it at events without worries.
8. As a gamer, I want a reward system for playing, to motivate me to play more.
9. As a gamer, I want to customize the sound of the machine, to create a gaming environment to my liking.

10. As a buyer, I want flexible payment options, to make purchasing the machine easier.
11. As a gamer, I want to see recommended machines, based on my preferences, to find what I like.
12. As a gamer, I want to see demo videos of each game, to decide which one to buy.
13. As a gamer, I want a multiplayer mode, so I can play with my friends.
14. As a buyer, I want warranty and after-sales service options, to feel secure with my purchase.
15. As a customer, I want an accessible customer support system, to resolve my queries quickly.

4. Minimum Required Functionalities

The application must include the following functionalities:

- Material Choice: Allow users to choose between wood, aluminum, or carbon fiber.
- Available Games List: Display the games available for purchase.
- Add Games: Include games in the machine to be purchased using a code.
- Finalize Purchase: Collect customer information for the delivery of the machine.

5. Additional Functionalities

In addition to the minimum functionalities, the following features are considered:

- Machine Customization: Allow users to choose the machine's color and the number of players (1 or 2).
- Game Category: View games by category and the category of games available on their machine.
- Game Reviews: Allow users to view and write reviews for games.

6. Object-Oriented Principles Analysis

The provided code implements a management system for arcade machines, utilizing object-oriented programming (OOP) principles. Below is an analysis of the main OOP principles applied in the code.

Encapsulation

Encapsulation is a fundamental OOP principle that restricts access to certain components of an object and exposes only what is necessary. In this code, the `User`, `Game`, and `ArcadeMachine` classes encapsulate their attributes and methods, providing clear interfaces for interacting with their data.

- **Classes and Attributes:** Each class has attributes specific to its functionality. For example, the `User` class has attributes like `id`, `name`, and `phone`, while `Game` has `game_id`, `title`, `category`, and `reviews`. These attributes are accessible through methods that validate and manipulate data, such as `validate_name` and `validate_phone` in the `User` class.
- **Private Methods:** Although the code does not explicitly use private methods, the use of methods like `validate_name` and `validate_phone` suggests an approach towards encapsulation, as these methods handle validation logic without directly exposing implementation details.

Inheritance

Inheritance allows new classes to be created based on existing classes, facilitating code reuse. In the provided code, no implementation of inheritance is observed, as all classes are independent. However, one might consider creating a base class to handle common attributes between `User`, `Game`, and `ArcadeMachine` if shared features are identified.

Polymorphism

Polymorphism allows different classes to implement methods with the same name, adapting to their specific contexts. In the code, the `to_dict()` method in the `Game` class allows a game object to be converted into a dictionary, which could be extended in other classes if they are implemented in the future. Although not explicitly used in this code, one can envision its use in a broader system.

Abstraction

Abstraction refers to the ability to represent complex concepts through simplified models. The `User`, `Game`, and `ArcadeMachine` classes represent specific concepts in the context of an arcade system, hiding the complexity of their internal implementation. For example, the `ArcadeMachine` class handles the logic of adding games and finalizing purchases, providing a clear interface for these operations without exposing the details of how data is stored or processed.

7. CRC Cards

CRC Card for the ArcadeMachine Class

ArcadeMachine	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Store information about the machine (material, color, number of players, available games).• Add games to the machine.• Finalize the purchase of the machine.	<ul style="list-style-type: none">• Game: Games are added to the machine.• User: Users make purchases of machines.

CRC Card for the Game Class

Game	
Responsibilities	Collaborators
<ul style="list-style-type: none">• Store information about the game (ID, title, category, reviews).• Add user reviews.• Convert game information to a dictionary format.• Save the list of games to a JSON file.	<ul style="list-style-type: none">• User: Users can leave reviews about the game.• ArcadeMachine: Games can be added to arcade machines.

CRC Card for the User Class

User	
Responsibilities	Collaborators
<ul style="list-style-type: none"> •Validate user name. •Validate user phone number. 	<ul style="list-style-type: none"> •Game: Can leave reviews about games.

8. Activity Diagrams

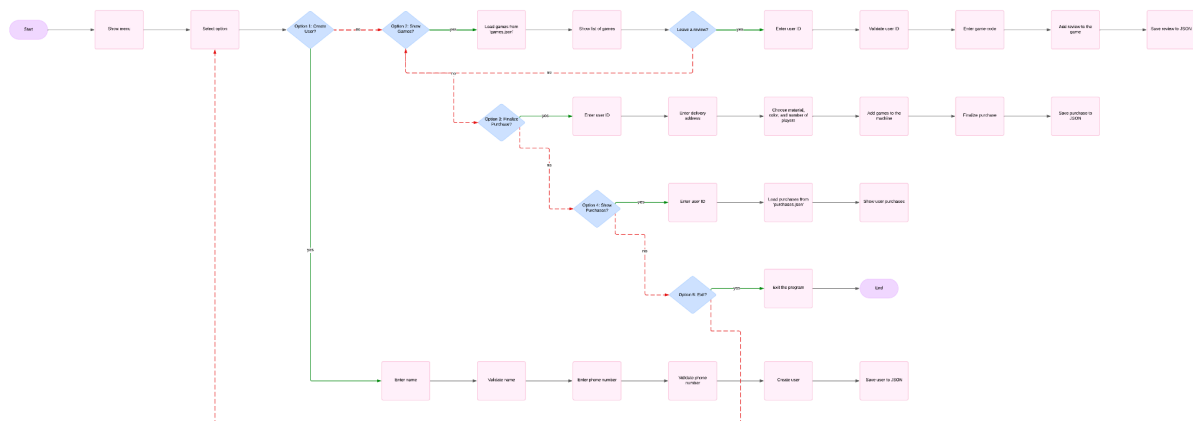
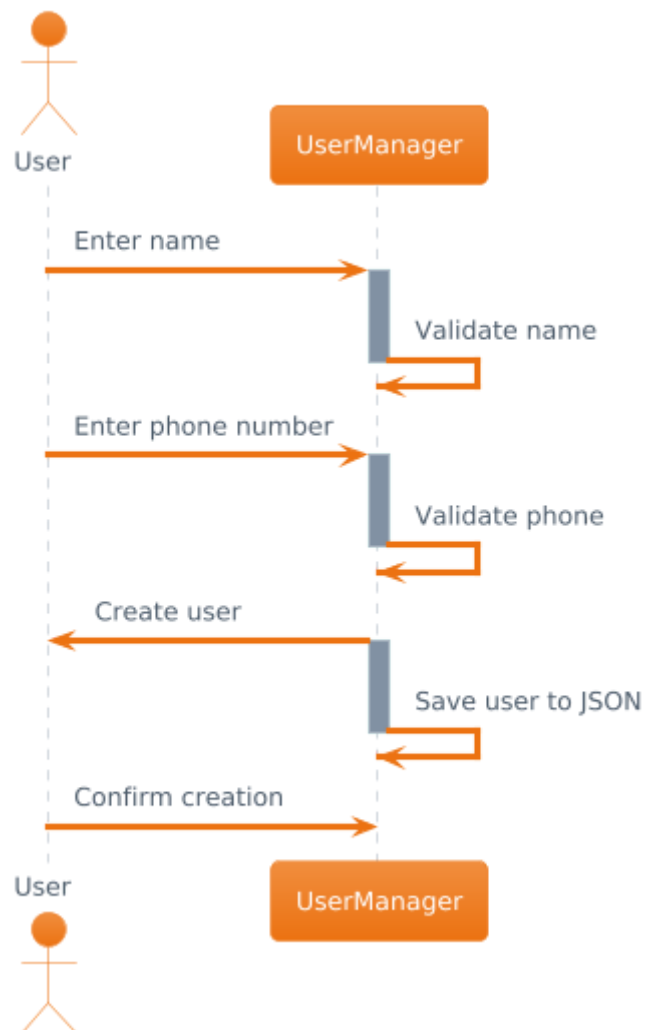


image link:

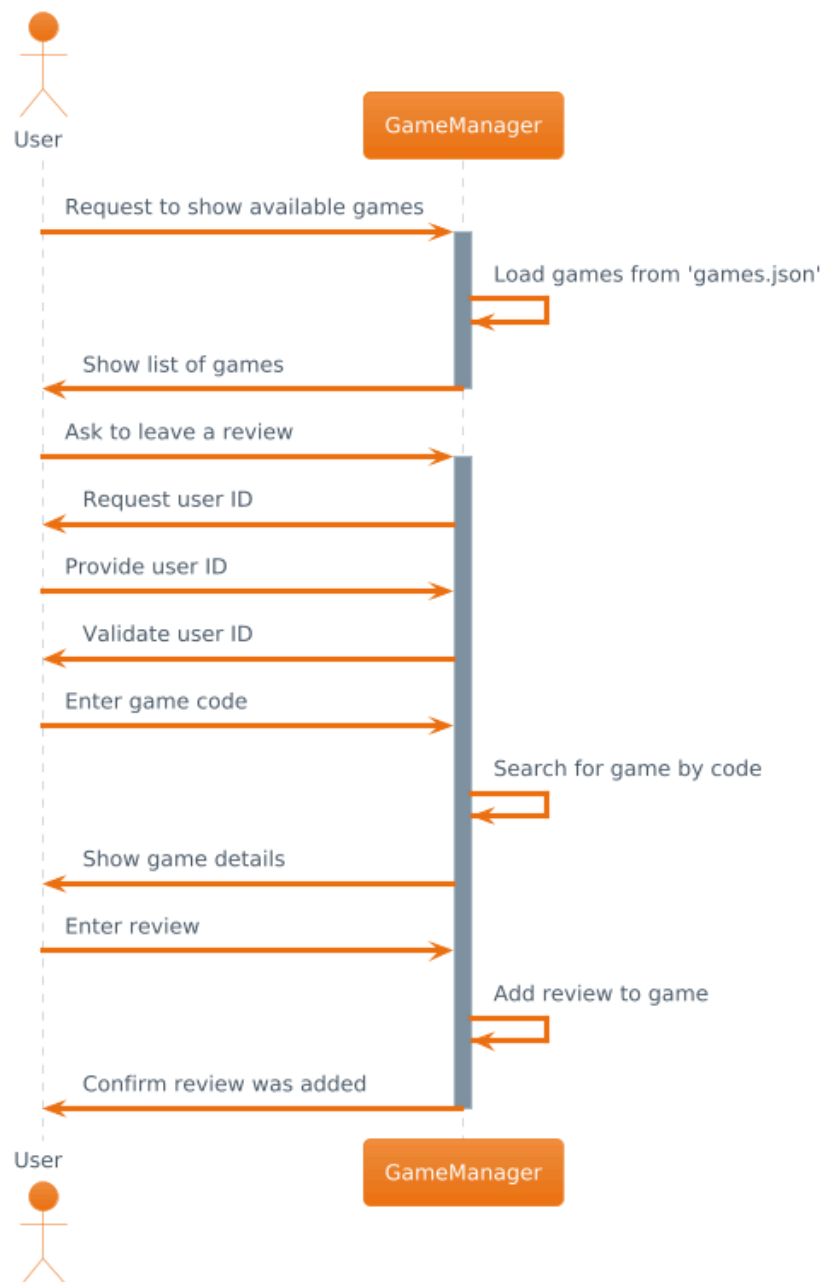
https://drive.google.com/file/d/1o1Dy6zbDwes1E8DhKTo5Kkr_aqGC0Pdh/view?usp=sharing

9. Sequence Diagrams

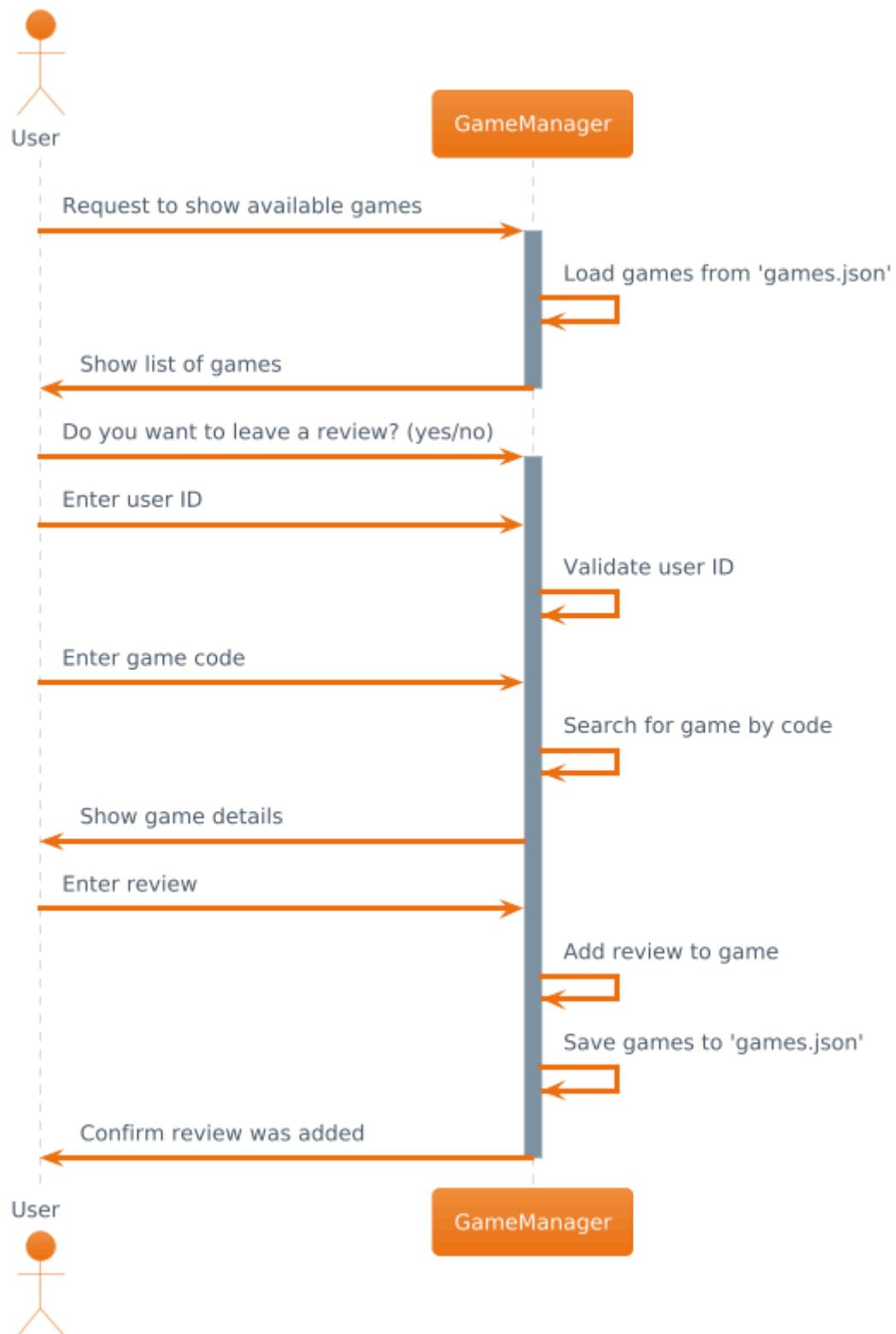
1. Sequence Diagram for Creating a User



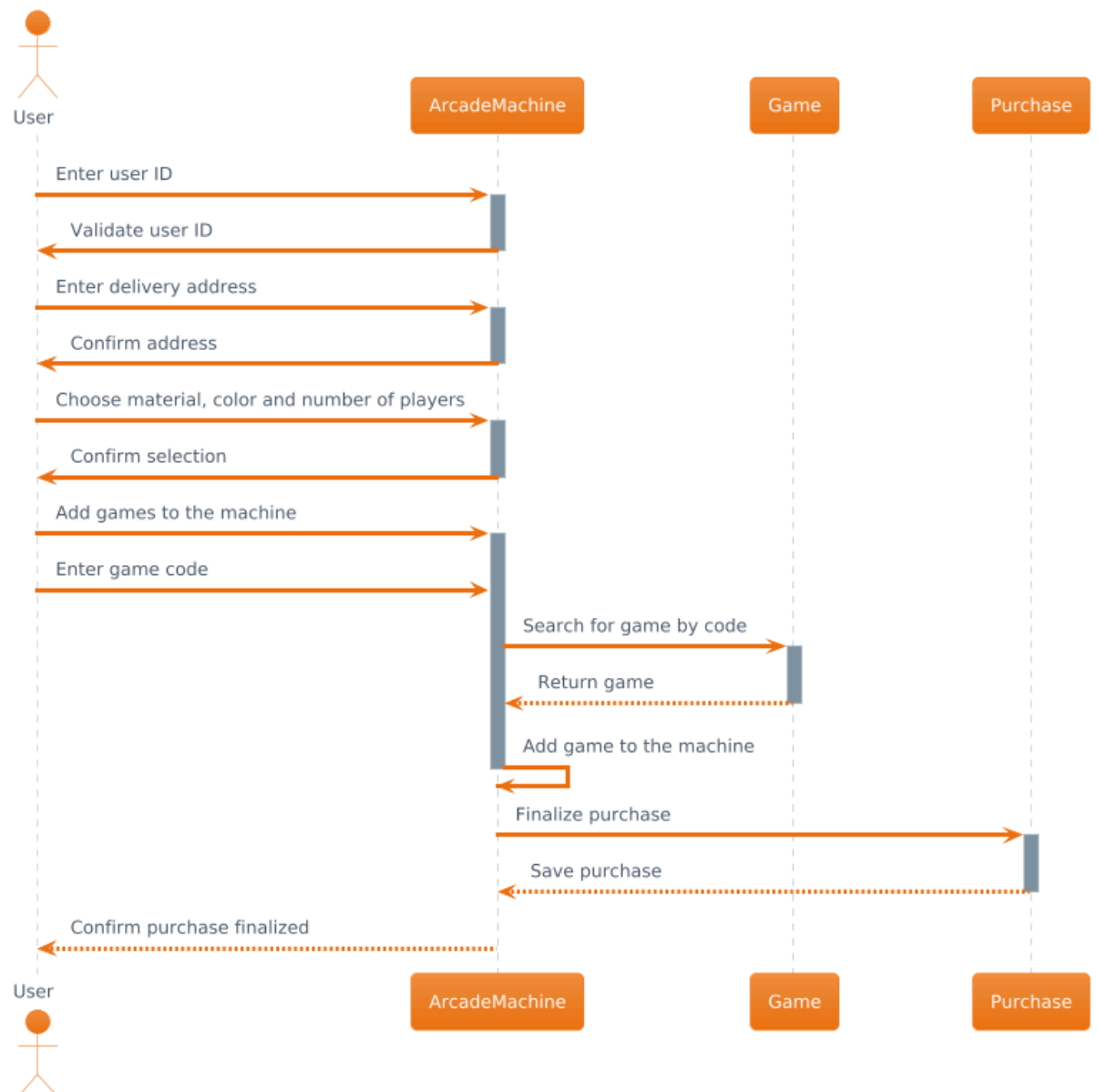
2. Sequence Diagram for Showing Available Games



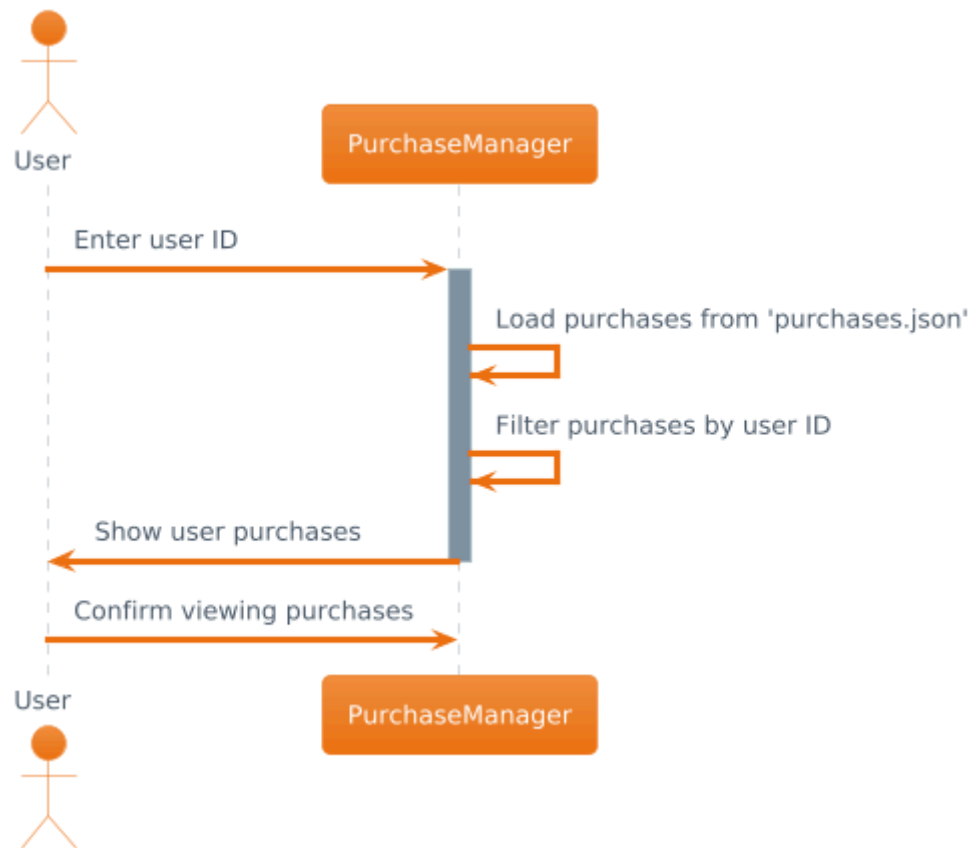
3. Sequence Diagram for Leaving a Review



4. Sequence Diagram for Finalizing a Purchase



5. Sequence Diagram for Showing User Purchases



10. Class Diagrams



11. Conclusions

The code adequately implements the principles of object-oriented programming, facilitating the management and manipulation of data related to users, games, and arcade machines. Although inheritance is not used in this context, encapsulation, abstraction, and the potential for polymorphism are present and are fundamental to the structure of the code. This modular approach not only improves the readability and maintainability of the code but also allows for future extensions and modifications with greater ease.

12. GitHub Repository

A GitHub repository will be created that contains all the documentation and code for the course workshops, organizing each workshop into its own folder, with a general README and specific README for each one. This report should be presented in PDF format and submitted before the established deadline.