

**Virtual Pet Adoption and Care Platform with Gaming and Social Features**

**GHULAM ISHAQ KHAN UNIVERSITY**

**CYBER SECUIRITY**

**SEMESTER 2 2023/2024**

**CS 112**

**Muhammad Abubakar 2023352**

**Muhammad Mehdi 2023465**

**Maisam Raza 2023463**

**Peerzada usman Ali qudsi 2023581**

**SUBMITTED TO:**

**PROFESSOR SALMAN SAEED**

**Documentation Guidelines**

**Introduction**

**Synopsis Project**

The project is a C++ program for a pet adoption and care platform, augmented with a gaming center and social features like adding friends and messaging. It provides users with functionalities to create accounts, add pets (dogs, cats, or parrots), remove pets, and interact with their pets through actions like feeding, grooming, playing, and resting. Additionally, users can buy accessories for their pets and play games like "Guess the Random Number" and "Rock-Paper-Scissors" to earn money.

The main function serves as the entry point of the program, presenting users with a menu-driven interface where they can select various actions. The program allows users to manage their pets, play games, add friends, and send messages to other users within the platform.

Overall, the project aims to simulate a virtual environment where users can care for their virtual pets, engage in entertaining activities, and interact socially with other users.

**Objective of The Project**

**Pet Management:** Enable users to create accounts, add pets (dogs, cats, or parrots), and manage their pets by performing actions like feeding, grooming, playing, and resting.

**Interactivity:** Facilitate interaction between users and their pets through a variety of actions that affect the pets' attributes such as energy level, cleanliness, and happiness.

**Gaming Center:** Provide users with entertaining games like "Guess the Random Number" and "Rock-Paper-Scissors" to earn virtual currency that can be used within the platform.

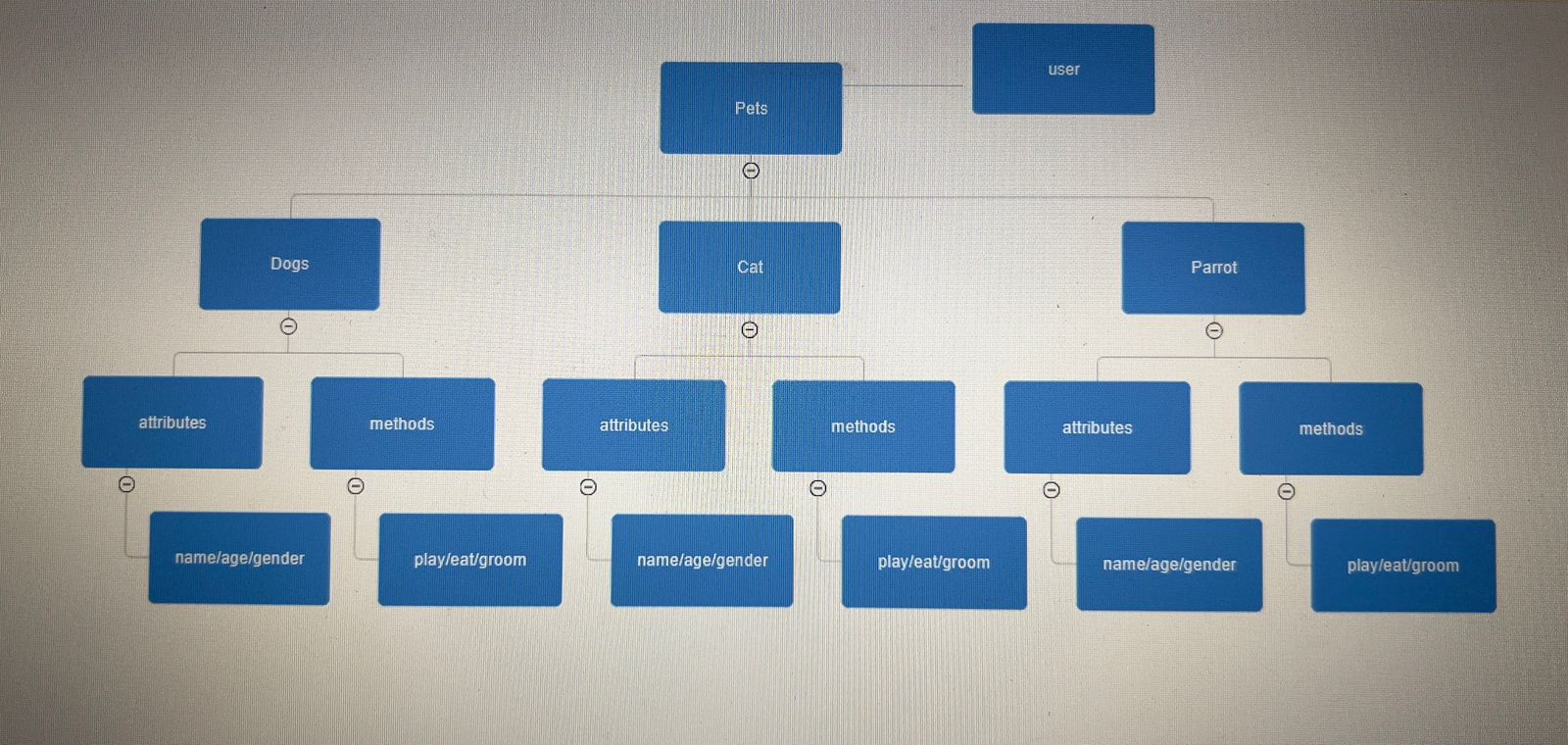
**Social Features:** Allow users to add friends, send messages, and engage in social interactions within the platform, enhancing the sense of community and camaraderie.

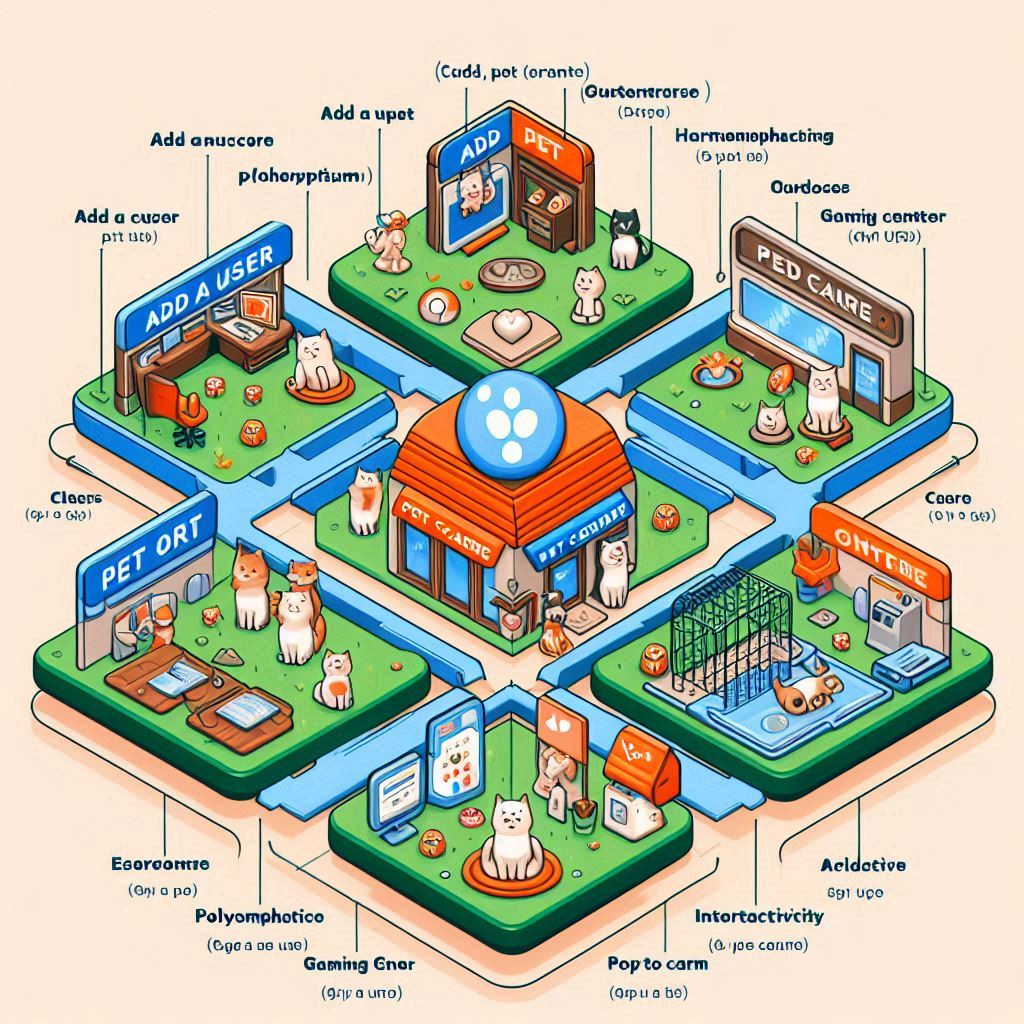
**User Experience**: Design a user-friendly interface with menu-driven navigation to ensure ease of use and accessibility for users of all skill levels

**System Design**

**2.1. UML Diagram**

A UML diagram is a way to visualize systems and software. Software engineers create UML diagrams to understand the designs, code architecture, and proposed implementation of complex software systems. UML diagrams are also used to model workflows and business processes. Below is the example:





**Used OOP Techniques.**

**Classes and Objects:** The program defines multiple classes such as pet, dog, cat, parrot, and user. Each class represents a real-world entity and encapsulates both data (attributes) and behaviors (methods).

**Inheritance:** The dog, cat, and parrot classes inherit from the pet class. This allows them to inherit common attributes and methods from the base class while also defining their own specific attributes and methods.

**Polymorphism:** Polymorphism is demonstrated through method overriding. For example, the getname() method is declared as virtual in the base class (pet) and overridden in the derived classes (dog, cat, parrot) to provide specialized behavior for each type of pet.

**Encapsulation:** Data encapsulation is achieved by declaring class attributes as protected or private and providing public methods (getters and setters) to access and modify these attributes. This ensures data integrity and allows for controlled access to class members.

**Abstraction:** Abstraction is utilized to hide the complex implementation details of the classes and provide a simplified interface for users. For example, users interact with pets through high-level actions like feeding, grooming, and playing, without needing to know the internal workings of these actions.

**Composition:** The user class contains vectors of dog, cat, and parrot objects, demonstrating composition by creating a "has-a" relationship between a user and their pets.

**Implementation:**

**Class Definitions:** Define classes such as pet, dog, cat, parrot, and user, along with their attributes and methods. These classes encapsulate the behavior and data of pets and users within the platform.

**Method Implementations**: Implement the methods defined in the classes to provide functionality such as feeding, grooming, playing, and messaging. This involves writing the logic to perform these actions and update the relevant attributes of pets and users.

**Main Function**: Write the main function to serve as the entry point of the program. The main function should provide a menu-driven interface for users to interact with the platform, select actions, and navigate through different features.

**User Interaction**: Implement user input handling to allow users to input their choices, such as selecting menu options, entering pet names, and sending messages to friends.

**Error Handling:** Include error handling mechanisms to validate user input, handle invalid options, and prevent runtime errors or crashes.

**Testing and Debugging:** Test the program extensively to ensure that all features work as expected and handle edge cases gracefully. Debug any issues encountered during testing to ensure the program's reliability and stability.

**Documentation:** Document the code thoroughly, including comments within the code to explain complex logic, clarify the purpose of methods and classes, and provide usage examples. Additionally, create external documentation such as a README file to guide users on how to use the program.

**1. Conclusion**

In conclusion, the C++ project successfully implements a virtual pet adoption and care platform enriched with gaming and social features. Through the utilization of Object-Oriented Programming (OOP) techniques, the project achieves modularity, extensibility, and maintainability, allowing for easy management and interaction with pets and users.

The project's main objectives include providing users with a comprehensive platform to care for virtual pets, engage in entertaining games, and interact socially with other users. By fulfilling these objectives, the project creates an immersive and enjoyable experience for users, fostering a sense of responsibility, entertainment, and community.

Through thorough testing and debugging, the project ensures reliability and stability, while comprehensive documentation aids users in understanding and utilizing the platform effectively

1. **Appendix (Code file)**

