**Quality Assurance Report**

Overview:

The Pet Adoption System’s code has been reviewed to evaluate its structure, functionality, and compliance with best practices. The system consists of several classes for handling pet management, volunteer coordination, events, and inquiries within an adoption center. Below is a detailed report highlighting its strengths. The code does not exist on github and was written purely for testing purposes. The testing technologies used include TSTL, unit tests, coverage, and universal mutator. These technologies come together to from a complete and cohesive test suite that can give valuable insights into the validity, correctness and reliability of the code.

Strengths:

Modular Design:

* The code is organized into separate classes, each responsible for a specific aspect of the adoption center management.
* The design encourages code reusability, scalability, and maintainability.

Clear Class Responsibilities:

* Each class has well-defined characteristics and methods that represent distinct functionalities.
* This improves code readability, making it easier to comprehend and maintain.

Error Handling:

* Some methods use error handling tools to deal with potential exceptions.
* This reduces runtime errors and increases system robustness.

Testing Process:

Code coverage



Unit tests



TSTL testing



Mutation testing



Conclusion:

Overall, the Pet Adoption System's code has a strong foundation because of its modular design, explicit class roles, and error handling features. The organised structure encourages code reuse, scalability, and maintainability, allowing for future upgrades or alterations. Furthermore, the thorough testing procedure, which includes code coverage analysis, unit tests, TSTL testing, and mutation testing, ensures the system's dependability and resilience.