Which one of the following requirements would be considered nonfunctional?
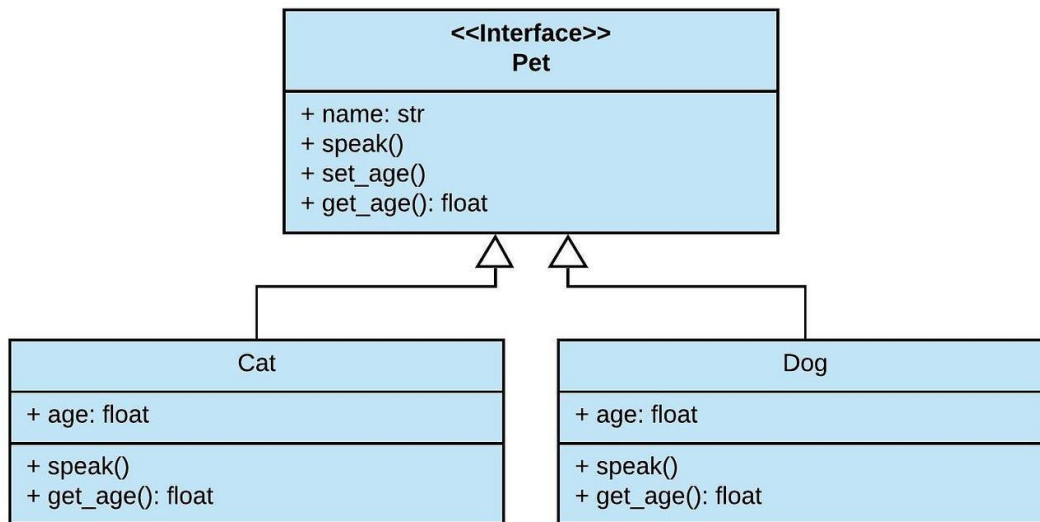
- A user should be able to sort through items on a website by type
- The system must allow users to sign up with a username and password to make an account
- The system should ensure users can log into the website within 2 seconds
- The system must be able to calculate costs, discounts, and savings in the shopping cart

Which of the following describes the functional requirements of a music streaming application?

- The system must provide a feature for users to rate songs on a star system from 0 to 5.
- The system must load playlists in under 5 seconds when selected by the user.
- The platform must support up to 15 million users concurrently using the system
- The system must transition to the next song within 2 seconds of clicking the skip button.

Why is it critical to perform thorough planning and analysis before moving to the design phase in a software development project?

- To ensure that both functional and nonfunctional requirements are comprehensively understood, enabling the design phase to address technical feasibility and user needs effectively.
- To prioritize user interface (UI) design early, allowing designers to integrate user feedback and refine visual elements before implementation begins.
- To allow the development team to create unit test cases in advance based on the initial analysis, aligning testing with system requirements.
- To identify potential bottlenecks in system performance during the design phase, ensuring architectural decisions are optimized for scalability.

```
            ┌──────────────────────────────────┐
            │         <<Interface>>            │
            │             Pet                  │
            ├──────────────────────────────────┤
            │  + name: str                     │
            │  + speak()                       │
            │  + set_age()                     │
            │  + get_age(): float              │
            └──────────────────────────────────┘
```

In the UML above, what is the relationship between Cat and Pet?

- Association
- Inheritance
- Aggregation
- Dependance

Which statement best describes the role of class diagrams in system design?

- Class diagrams focus primarily on modeling user interface elements and interactions.
- Class diagrams represent object-oriented components, showing their attributes, behaviors (methods), and the relationships between different classes in the system.
- Class diagrams are used exclusively to model the structure of a relational database, including tables and relationships.
- Class diagrams define the overall system architecture and illustrate how data flows between components.

How does the design phase ensure alignment with both functional and non-functional requirements in software development?

- By prioritizing the creation of a user interface early in the design process.
- By focusing solely on coding practices and unit testing to ensure correct functionality.
- By considering both the system's architecture and the detailed design of components to address specified functional and non-functional requirements.
- By testing the software's user experience to ensure usability and satisfaction.

Which of the following methods would be most helpful when starting to code based on requirements?

- Methods that bypass user inputs to focus on internal logic.
- Methods that return a random value to test the system's response to unexpected outputs.
- Methods that only handle errors to prevent crashes during execution.
- Methods that check system constraints and ensure the expected inputs and outputs are validated against the requirements.

When implementing methods, how does adhering to the UML class diagram help ensure the code is correct?

- It assists in automating code generation, minimizing manual coding efforts.
- It eliminates the need for testing by guaranteeing that the code is inherently correct.
- It ensures that the methods align with the system's intended structure and relationships, promoting consistency and correctness in implementation.
- It simplifies the user interface design by providing clear guidelines on visual elements.

Test Class

```python
class TestCat(unittest.TestCase):
    def test_initialization(self):
        cat = Cat(3.5)
        self.assertEqual(cat.age, 3.5)
    def test_speak(self):
        cat = Cat(2.0)
        self.assertEqual(cat.speak(), "Woof")
    def test_get_age(self):
        cat = Cat(4.0)
        self.assertAlmostEqual(cat.get_age(), 4.0)
```

In the above test class snippet, what is the issue with the given test classes for the Cat class?

- The Cat class does not have a __str__ method, which causes the tests to fail
- The test_get_age method has an incorrect expected value in the assertion.
- The test_speak method has an incorrect assertion for the return value of the speak method.
- The test_initialization method has incorrect assertions for the age field.

What should be the focus of unit testing during the testing phase?

- Validating that individual components or methods function correctly.
- Focusing solely on performance optimization and load testing
- Ensuring that the entire system works as intended from end to end.
- Testing the integration of multiple components within the system.

How can you determine whether your unit tests are comprehensive enough?

- If the system passes a few basic test cases, indicating some level of functionality.
- If the user interface appears visually correct, indicating that the implementation aligns with design specifications.
- If the code compiles without syntax errors, suggesting it's correctly written.
- If the tests comprehensively cover normal cases, edge cases, and unexpected inputs, ensuring that all scenarios are tested without failure.