Which of the following is an example of a functional requirement for an online music website?

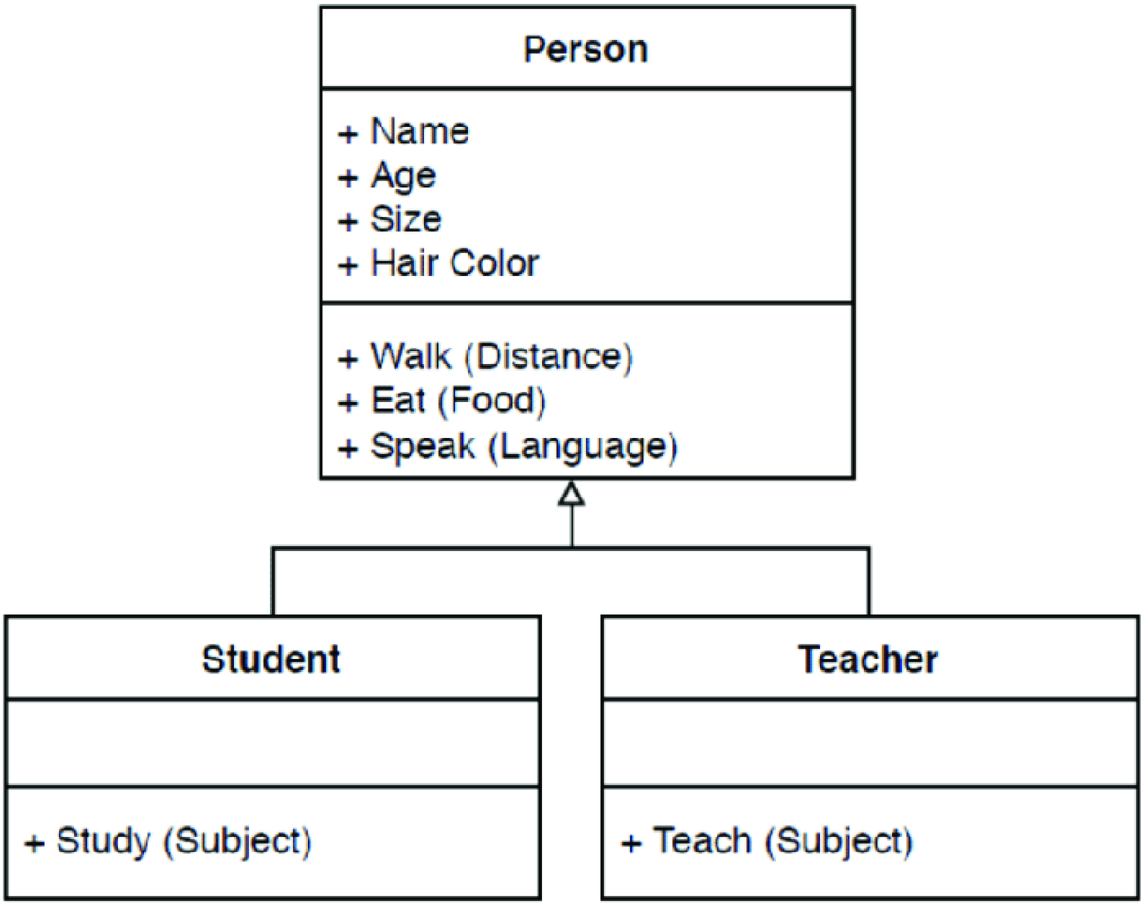
- Users can create playlists by selecting songs from the library.
- The website should load within 2 seconds on all modern browsers.
- The interface should be aesthetically pleasing and user-friendly.
- The website must support mobile and tablet views.

Which one of the following requirements would be considered nonfunctional?

- A system user should be able to add something to a shopping cart
- A system user should be able to send items to their shopping cart within 10 seconds
- A user should be able to log in using a username and a password
- A user should be able to sort through items to add to the shopping cart by item type

Why is the requirement phase crucial before moving into the design phase of the software development lifecycle?

- It provides a clear understanding of user needs, ensuring the design aligns with the system's objectives.
- It allows developers to start coding immediately without needing additional input.
- It identifies all potential technical challenges and ensures they are addressed early in the process.
- It eliminates the possibility of costly changes by giving all functions and helper functions upfront.



Using the UML above, what is the relationship between Teacher and Person?

- Aggregation
- Dependance
- Inheritance
- Association

What role do class diagrams NOT play in system design?

- Representing the relationships and interactions between classes.
- Defining the way information flows through classes
- Depicting the system’s behavior over time.
- Serving as a blueprint for implementing classes and their functions.

How does the design phase follow the requirements created in software development?

- By translating the documented requirements into a structured blueprint for implementation.
- By ignoring non-functional requirements focus solely on the system's architecture.
- By directly implementing the features outlined in the requirements document without further planning.
- By finalizing the requirements based on assumptions rather than user input.

What is the advantage of implementing methods that validate functional requirements established during the requirement phase?

- It avoids having to test the code later
- It helps improve the aesthetics of the code
- It ensures that the implementation matches the intended functionality
- It allows developers to skip the design phase

How can a method be implemented to ensure compliance with a UML class diagram?

- By focusing on the user interface and ensuring it aligns with the design goals.
- By matching the method names, return types, parameters, and relationships as outlined in the UML class diagram.
- By creating methods as needed to address emerging requirements during development.
- By writing methods that handle exceptions to ensure system stability.

Test Class

```
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(), "Meow")

    def test_get_age(self):

        cat = Cat(4.0)

        self.assertAlmostEqual(cat.get_age(), 5.0)
```

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

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

What is the next step after writing code to implement classes and methods in the software development lifecycle?

- Ignoring further testing
- Refactoring the design phase
- Testing and ensuring that the methods meet the requirements and function as expected
- Changing the UML class diagram

Why is it important to cover edge cases in unit testing?

- To improve the performance of the system by optimizing the code paths.
- To minimize the need for detailed documentation by validating the implementation through tests.
- To ensure that the methods correctly handle possible inputs, including unexpected, extreme, or boundary cases, thus enhancing reliability.
- To eliminate the need for integration testing by ensuring all components work correctly in isolation.