Week 2 Assignment

Miguel Larios

CST499: Capstone for Computer Software Technology

Dr. Butler

11/20/2023

**Testing**

Software testing is a critical phase of software engineering, ensuring the functionality and reliability of a software product. It includes multiple phases, each focusing on different objectives, such as component testing, integration testing, system testing, and acceptance testing. These multiple testing phases ensure the software operates as expected and help mitigate issues early before releasing the product.

Component testing is an important phase in testing that focuses on testing the components at the individual level. The usability and functionality of each component is independently tested in an isolated and controllable state. This phase helps detect issues early on. Conducting component testing at the individual level helps ensure each component operates as it should, which also helps in later testing phases (“Component Software Testing,” 2019).

Integration testing is a test of interactions between components within a system. This phase normally focuses on the functional aspects of the system, evaluating how the integrated components work together to ensure a seamless operation. There are various approaches to integration testing, such as the top-down approach, which tests higher-level modules first, the bottom-up approach, which starts testing lower-level modules initially, the big bang approach, where all modules are tested simultaneously, and hybrid approaches that combine elements of both top-down and bottom-up methodologies (Rana, 2019).

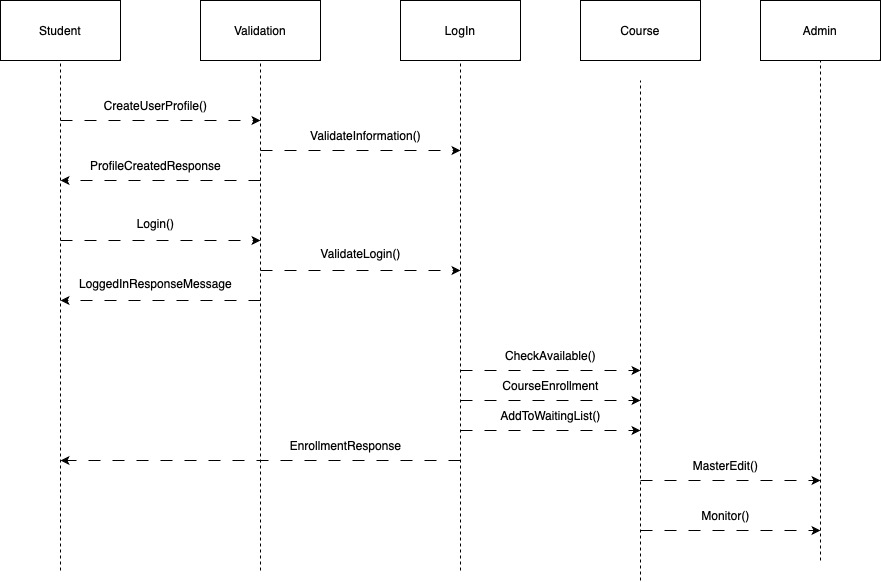
System testing is a test that reviews the entire system. It assesses it functionalities and performance as a whole. It is typically conducted after integration testing, focusing on ensuring it functions according to the requirements and specifications. It differs from the other testing levels as it goes beyond testing individual components. System testing evaluates non-functional and functional items to ensure they meet all expectations before deployment (Rana, 2019).

Acceptance testing is a phase where the customers test the software before formally accepting the product. The customer tests are based on the criteria that are established during the requirements phase. The purpose is to ensure the software is adhering to the established specifications. Typically, test cases are outlined to ensure they meet the original expectations (Tsui et al., 2018).

UML diagrams are a great way to visualize a system and behaviors. Which also makes them great for testing. An example is the System testing phase, which utilizes a sequence diagram to represent system testing results, similar to Figure 1. A sequence diagram should show the expected action or behaviors. Having a visual of the expected behaviors serves as a baseline that can be compared to the test behaviors to ensure they align and work as designed (Nayak & Samanta, 2012).

**Figure 1**

Sequence Diagram

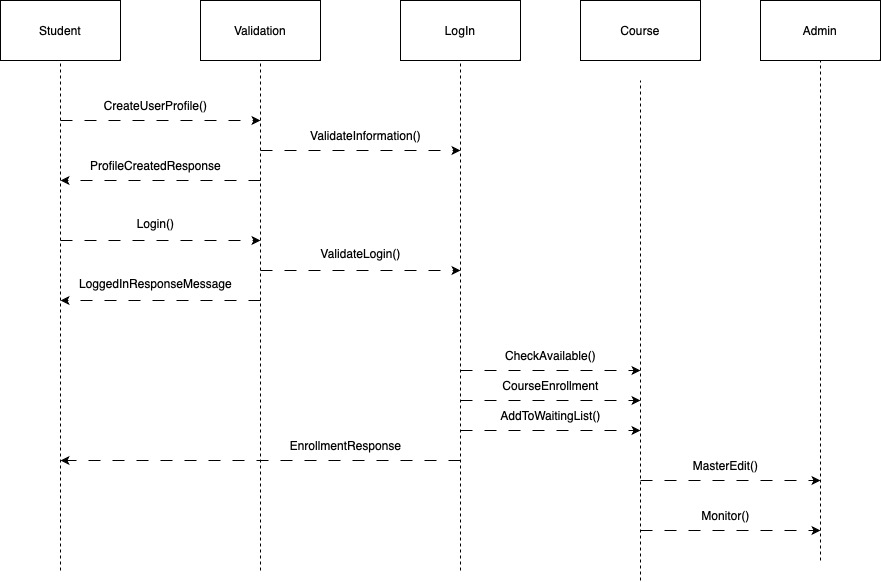


Software testing is a critical step in the software engineering life cycle. It ensures that the software behaves as expected and that the client is provided with a product that meets their expectations. The various testing phases, component, integration, system, and acceptance, are essential to test every aspect of the system. Component testing tests at the individual level, integration tests for component interactions, system testing evaluates the system as a whole, and acceptance testing is the final test that customers conduct to ensure the product behaves as expected. This framework ensures adherence to the requirements and helps a team provide dependable and efficient products.

UML Diagrams

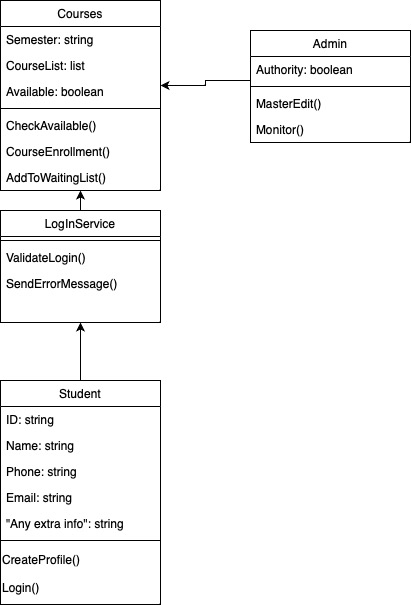
**Figure 1**

Sequence Diagram



**Figure 2**

Class Diagram



References

Component software testing. (2019, August 14). *GeeksforGeeks*. https://www.geeksforgeeks.org/component-software-testing/

Nayak, A., & Samanta, D. (2012). Synthesis of test scenarios using UML sequence diagrams. *ISRN Software Engineering*, *2012*, 1–22. https://doi.org/10.5402/2012/324054

Rana, K. (2019, December 6). *System testing vs integration testing: Key differences*. ArtOfTesting. https://artoftesting.com/difference-between-system-and-integration-testing

Tsui, F. F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones & Bartlett Learning.