Week 2 Assignment UML Design Modeling

Thomas Pedersen

CST499 Software Technology Capstone

Professor Charmelia Butler

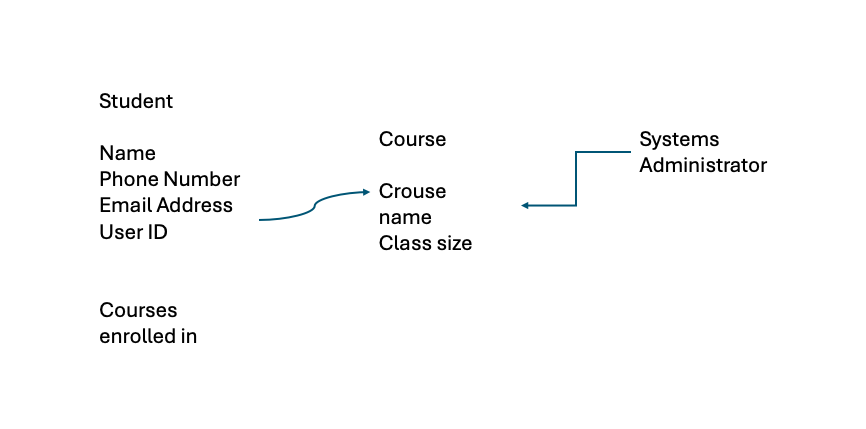
November 18, 2024

(Geeks for Geeks, 2024)

**UML MODELS**

Unified Modeling Language (UML) models are visual representations of software systems and designs used in software engineering. These models facilitate the organization, communication, and documentation of system components, including their structure, behavior, and relationships. Common UML models include class diagrams, use case diagrams, sequence diagrams, state diagrams, and activity diagrams. According to Geeks for Geeks, “It is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language.” (Geeks for Geeks, 2024) I like how they explain that it's a blueprint for programming. I think it's your blueprint for your house.

Class Diagram Figure 1



Sequence Diagram Figure 2

A diagram of a course

Description automatically generated

Use Case Diagram Figure 3A diagram of a diagram

Description automatically generated

System testing is a software testing methodology that assesses the functionality of an integrated IT system to ensure compliance with predefined requirements. It is typically executed following integration testing and preceding acceptance testing. System testing encompasses the evaluation of the system’s overall behavior and the seamless interaction between its constituent components. Additionally, it verifies that the system adheres to both business and technical specifications. The testing process is designed to identify errors in component interactions and discrepancies between anticipated and actual outcomes. System testing is commonly conducted by a collaborative team comprising testers, developers, and end-users, who can provide valuable feedback based on their respective perspectives. According to Sophia Iroegbu, “In system testing, the quality assurance team evaluates how each component of the application or software work together in a full, integrated environment.” (Iroegbu, 2022) Sophia provided a comprehensive explanation of the software development process, emphasizing the collaborative nature of teams in ensuring its seamless operation.

Integration testing is a phase in software testing that involves testing the interfaces and interactions between individual components or modules of a software system. Its goal is to identify the integration errors and ensure that the components work together as expected. Integration testing can be performed at different levels, such as unit integration, system integration, and end-to-end integration.

The purpose of integration testing is to find errors in the interfaces between components, which can be difficult to detect during unit testing when the components are tested in isolation. Integration testing is usually conducted by testers, developers, and sometimes end-users.

Acceptance testing, also known as user acceptance testing (UAT) or beta testing, is a software testing methodology performed by end users or clients to validate the software’s compliance with their requirements and specifications. The primary objective of acceptance testing is to ascertain whether the software is suitable for deployment or live operations. During acceptance testing, users simulate real-world scenarios to ensure the software’s functionality, alignment with business objectives, and adherence to functional specifications documented in the functional specifications document. Typically, acceptance testing occurs following the integration and system testing phases.

Component testing, or unit testing, is a software testing methodology that involves testing individual components or units of a software system in isolation to ascertain their intended functionality. Typically performed by the development team, component testing focuses on testing small, self-contained functionalities. The primary objective of component testing is to identify and rectify errors or defects in the codebase at an early stage of the development process, prior to integration with other components. This approach mitigates risks associated with integration and system testing, thereby enhancing the overall quality of the developed software.

# Works Cited

Geeks for Geeks. (2024, Oct 23). *Unified Modeling Language (UML) Diagrams*. Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/

Iroegbu, S. (2022, Sep 21). *What is Software Testing? A Beginner's Guide*. Retrieved from freeCodeCamp: https://www.freecodecamp.org/news/software-testing-beginners-guide/