UML Design Modeling

Dallas Spencer

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Instructor: Charmelia Butler

December 29th , 2024

An important part of any software is having a plan in place for testing and development. In this essay we will be going through the different levels of testing and showcasing the requirements for the software in UML charts.

In the software development life cycle, testing can and should be completed at every stage of the process. Each phase of testing contains what is called a test case. These test cases are used to stress the system on pretend scenarios that either attempt to force a fault or prove a function works. These test cases are created before the modules are completed, following a test-driven development process.

Before diving into the testing phases, it's important to understand what black-box and white-box testing are. Black box testing is a method of testing that doesn’t care about the inner workings of the test subject. It focuses on the input and getting the expected output (PractiTest, 2024). White box testing is the opposite, this method of testing focuses on the inner workings of the test subject. Both are relevant and equally important to the testing process; each phase of testing can be defined by one of these or a combination of both.

Testing begins on an individual level with a white box style of testing, component testing. Component testing is testing individual modules, making sure they function on their own. This means testing each individual line of code within the module, running test scenarios, and developing ensuring the specific module meets the clients demands.

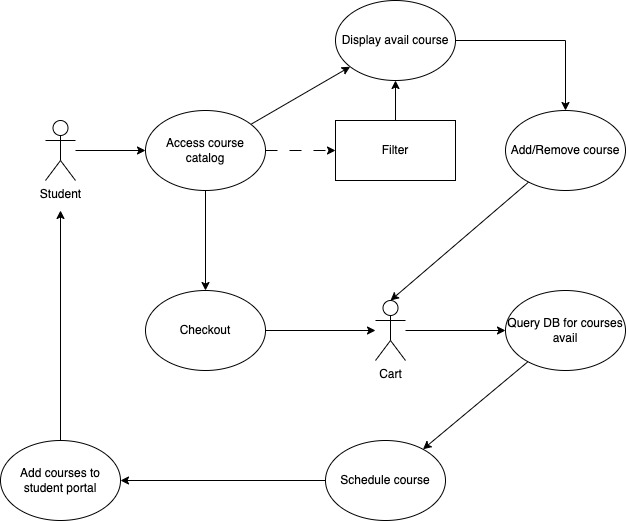
Once component testing (often called unit testing) is complete, we begin integration testing, another form of white box testing. This takes two or more tested modules from the prior step and begin testing their integration together (Geeksforgeeks 2024). In these tests they go through verification that they are able to work as expected and allow for any errors to be corrected between modules.

Once it's confirmed that individual models communicate with each other as expected, the following step is system testing, a form of black box testing. System testing takes all tested components from unit tests and component tests and tests it. The primary goal of system testing is detecting defects between integrated units and the system as a whole (Yasar et al., 2023).

Finally, once system testing is completed, we move on to acceptance testing, a form of black box testing. These tests are specific to ensuring that the requirements that were met are set by the stakeholders in the requirements gathering phases. Acceptance testing can take many forms, from third party vendors to beta testing with end-users. The method of effective acceptance testing depends on the stakeholders and the goal of the project.

In conclusion, testing is an integral process to developing software. It begins with a focused view at module testing and as the software grows, so does the scope of testing. Without effective testing there is no way a project will be able to meet its quality standards that are expected by the stakeholders.

UML Diagrams for the software components:

  
  
Figure 1: Use case diagram for course catalog

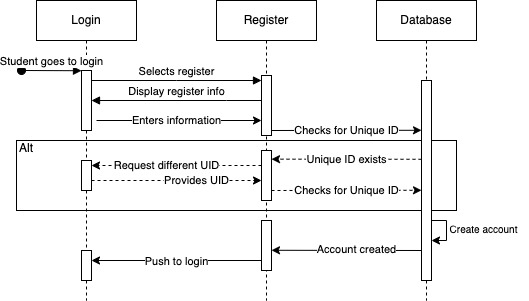


Figure 2: Sequence diagram for register function

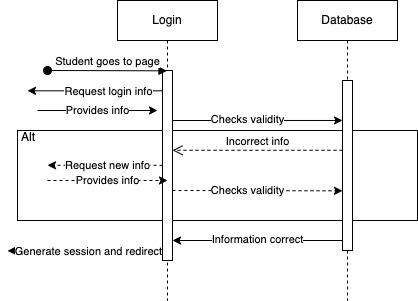


Figure 3: Sequence diagram for login function

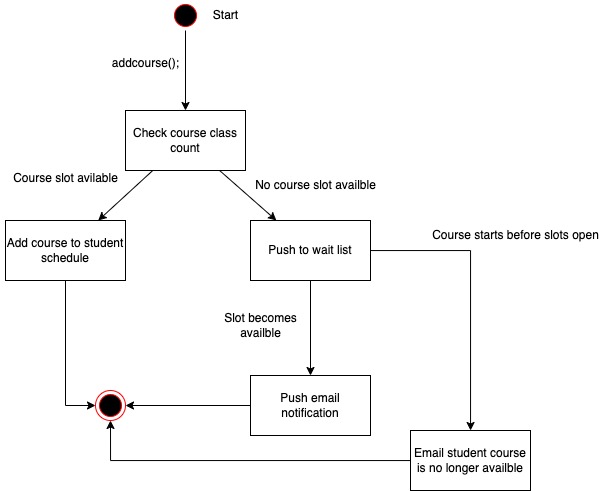


Figure 4: State diagram for wait list function

Resources:

*Black Box vs White box testing | PractiTest*. (2024, September 19). PractiTest. <https://www.practitest.com/resource-center/article/black-box-vs-white-box-testing/>

GeeksforGeeks. (2024, February 25). *Levels of software testing*. GeeksforGeeks. <https://www.geeksforgeeks.org/levels-of-software-testing/>

Yasar, K., & Black, R. (2023, March 14). *system testing*. Search Software Quality. <https://www.techtarget.com/searchsoftwarequality/definition/system-testing>