UML Design Modeling and Testing

Jacob Colburn

University of Arizona Global Campus

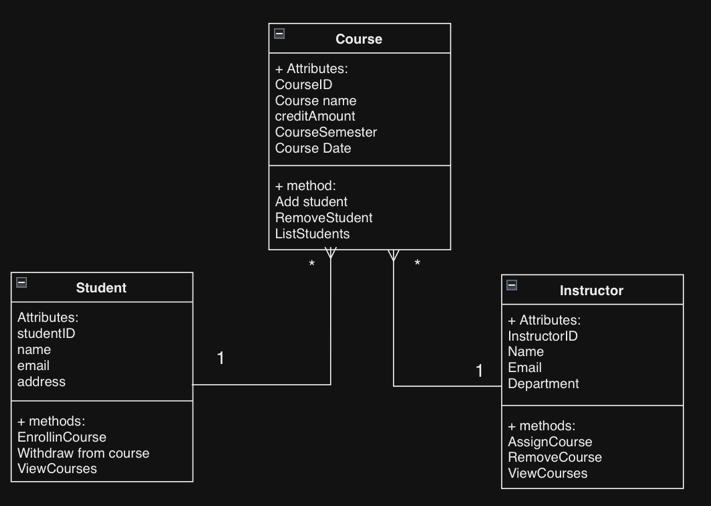
CST499: Capstone for Computer Software Technology

Instructor: Prof. Joseph Rangitsch

05AUG24

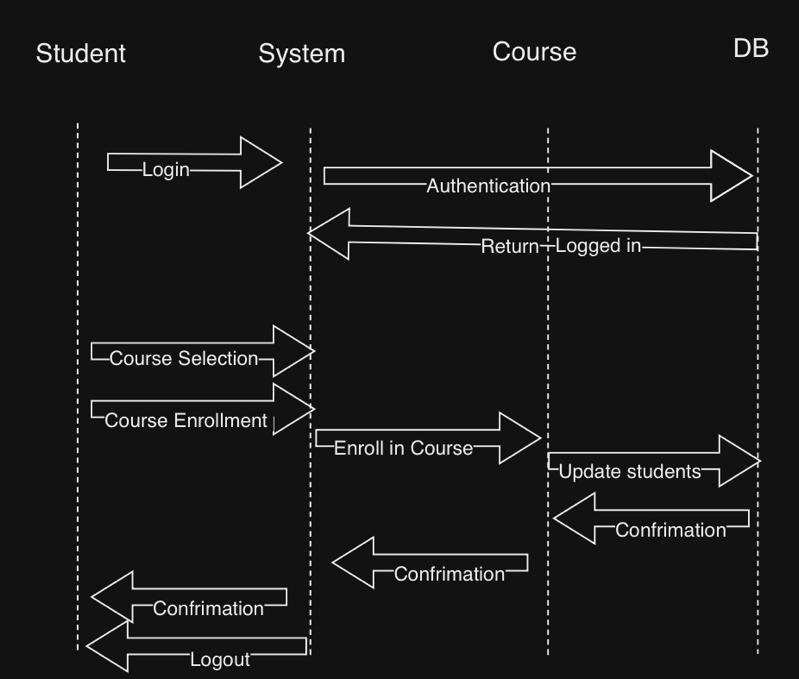
**Introduction**

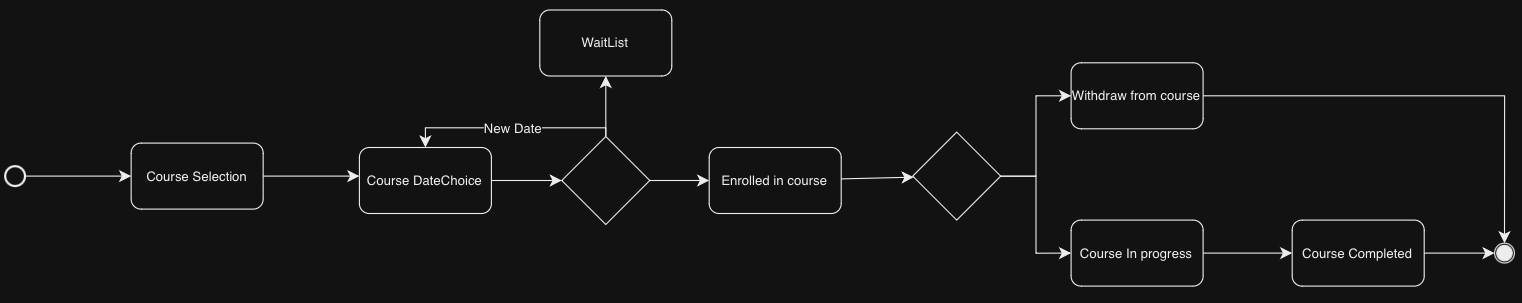
Software testing is an important part of the development process, ensuring the systems quality and reliability meet the expectations of stakeholders. UML models help to guide the testing phase by illustrating the interrelations, behavior, and structure of the system (Aldaeej, A., & Badreddin, O., 2016). In this assignment, we will look at the different levels of software testing and demonstrate how UML models can support and enhance the testing process.

**Component Testing**

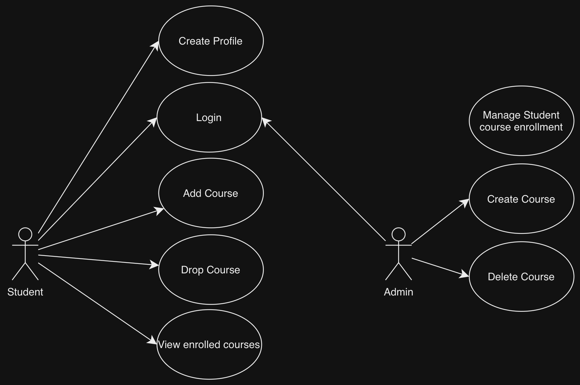
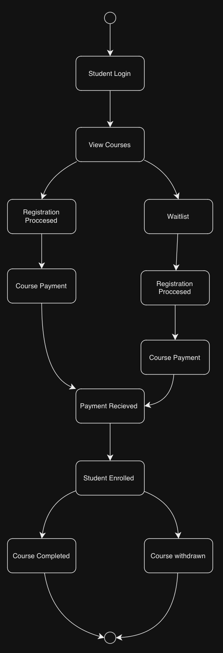
Component testing (unit testing) is used to ensure all functionality of any individual classes or components in the systems (Tsui, F., Karam, O., & Bernal, B., 2018). During Component testing, the class diagram is referenced to provide the details of those components such as the attributes, methods, and relationships. If component tests are implemented for each of the component's, developers can ensure that all of those components function as intended. As an example, developers could test the student class to ensure the methods of courseEnroll and courseWithdraw function as intended.

**Integration Testing**

 Integration testing tests the interactions between the components in the system to ensure functionality. Sequence diagrams are a great help in this testing phase by illustrating the flow and interactions between the objects in the system. Verifying the integrated components can communicate and work together as intended is the purpose of this testing phase (Madhurri, A., 2024). An example of integration is testing interactions between student and course objects when the student enrolls in a course.

**System Testing**

System testing tests the entire completed and integrated system, validating any specified requirements. Using an activity diagram, the workflow of the system is illustrated, guiding any testers through the processes of the systems functionality (Rapolu, K., 2018). As an example, the entire system would be tested from start to finish to ensure that students can properly create an account, create a profile, update the profile, and enroll/withdraw/waitlist for courses.

**Acceptance Testing**

Acceptance testing verifies the system meets the requirements that were set for the product to ensure its ready for deployment. State and use case diagrams are important in this phase of testing. State diagrams illustrate the different states an object can be in and how it transitions from one to the other. Use case diagrams are used to outline any possible interactions of the actors and system (Rapolu, K., 2018). An example of these diagrams would be states as in enrolled, failed, passed, withdrawn, and use cases from actors would be logging in, enrolling in courses, withdrawing from courses, etc.

**Conclusion**

UML models are a crucial part of the software testing model and integrating them ensures a more comprehensive approach to high quality software. Visualizing the use cases, structure, classes, and behaviors of the system through the UML diagrams help developers and testers the tools to execute testing effectively on all testing levels. This approach ensures that developers can identify and fix any problems with the system earlier, leading to a reliable, robust, and maintainable system

**References**

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning. [Course Text](https://platform.virdocs.com/read/2348054/11/#/4/2[ch06]/6/4[ch06-sect1-002]/10/4/2)

Aldaeej, A., & Badreddin, O. (2016, May 14–22). [Towards promoting design and UML modeling practices in the open source communityLinks to an external site.](https://go.openathens.net/redirector/ashford.edu?url=https%3a%2f%2fieeexplore.ieee.org%2fdocument%2f7883384%3farnumber%3d7883384%26SID%3dEBSCO%3aedseee) [Paper presentation]. 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), Austin, TX, United States. <https://doi.org/10.1145/2889160.2892649>

Madhurri, A., (2024). “Levels of Software Testing”. <https://www.geeksforgeeks.org/levels-of-software-testing/>

Rapolu, K., (2018). “Selection of UML Models for Test Case Generation: A Discussion on Techniques to Generate Test Cases”. St. Cloud State University. https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1031&context=csit\_etds