**Week 2 – Assignment**

Chandler DeYoung

University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Joseph Rangitsch

May 27th 2025

In order to create the proposed project, it is important to outline everything involved in the project in UML diagrams. These diagrams will provide developers with the tools and direction needed to create the overall system and lead to a successful deployment. On top of that, understanding the different testing types is key to limiting the amount of bugs in the code and providing a great user experience on launch.

This first diagram is a class diagram of the static structure of the system. This provides the table configurations and how they interact with each other.

*A computer screen shot of a computer

AI-generated content may be incorrect.*

The next diagram is a use case diagram that shows the system’s interactions with the outside users like students, instructors, admins, etc.

A diagram of a diagram

AI-generated content may be incorrect.

Up next for the diagrams is a sequence diagram that shows the process a student would go through to register for a class, and if they get put on the waitlist.

A diagram of a student

AI-generated content may be incorrect.

This diagram will show the workflow of the course enrollment process in an activity diagram.

A screenshot of a computer screen

AI-generated content may be incorrect.

Last up is going to be the state diagram showing off the state transitions during the process.

A diagram of a student

AI-generated content may be incorrect.

As far as testing goes, for component testing, each piece of the software like the user login, course enrollment, and waitlist will be tested individually in order to make sure everything functions as intended. Unit tests will be done to make sure the data is correct after all the processing and interactions are completed in the system.

Integration testing will be done after all the individual pieces have gone through component testing. As an example of the multiple pieces being tested together, once a student has been logged in, the functionality of viewing and enrolling in courses will be tested. Integration testing will help make sure that data is able to flow properly between the user interface, the main sequence flows, and the database.

System testing is about making sure the entire course registration system works as a complete package. This would involve doing things like simulating real life scenarios like multiple people registering at the same time, dealing with session timeouts, and ensuring the database is still accurate during all the enrollments and unenrollment.

Lastly, acceptance testing will be done to make sure that the system matches and hits all the requirements and meets the stakeholder’s expectations. Then a small group of actual users such as students and teachers will do tests based on some of these real life scenarios. If the acceptance testing is met with good feedback, then that will finally say the system is ready for deployment.

Overall, the UML diagrams will help provide a blueprint of the system and it’s behaviors. By combining all the diagrams, an in depth view of the system will come into picture. On top of the diagrams, it’s important to know all the testing types and when is the right time to perform all those tests. All of this efforts is done to create a seamless registration process for everyone involved.

References:

Dooley, J. F. (2017). *Software development, design, and coding: With patterns, debugging, unit testing, and refactoring. (2nd ed.).* Springer Nature. <https://ebookcentral.proquest.com/lib/ashford-ebooks/reader.action?docID=5162983>

GeeksforGeeks. (2025, January 2). *Unified modeling language (UML) diagrams*. GeeksforGeeks. https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/